



A MATEMÁTICA POR TRÁS DAS MANDALAS

Luciane Ferreira Alcoforado¹

João Paulo Martins dos Santos²

Resumo

Este trabalho oferece uma exposição artística que combina a beleza das mandalas com a precisão da matemática. As mandalas são criadas a partir de uma variedade de conceitos matemáticos, incluindo expressões analíticas de curvas paramétricas, homotetias e isometrias. Utilizando a representação no plano cartesiano, cada mandala é meticulosamente projetada para incorporar elementos matemáticos que resultam em padrões complexos. Além disso, o processo de coloração das mandalas é realizado de diversas formas, entre estas destaca-se a seleção aleatória de cores, proporcionando uma variedade de combinações de cores, contribuindo para a singularidade de cada mandala. Para a produção e coloração dos desenhos, é empregada a linguagem de programação R, oferecendo uma abordagem computacional eficiente e flexível. Isso não apenas facilita o processo de criação, mas também permite a experimentação e a personalização das mandalas de acordo com as preferências do criador.

Palavras-chave: Matemática, Mandalas, Arte, Linguagem R.

Abstract

This work presents an artistic exhibition that combines the beauty of mandalas with the precision of mathematics. The mandalas are created from a variety of mathematical concepts, including analytical expressions of parametric curves, homotheties, and isometries. Using representation in the Cartesian plane, each mandala is meticulously designed to incorporate mathematical elements that result in complex patterns. Furthermore, the process of coloring the mandalas is carried out in various ways, among which the random selection of colors stands out, providing a variety of color combinations that contribute to the uniqueness of each mandala. For the production and coloring of the designs, the R programming language is employed, offering an efficient and flexible computational approach. This not only facilitates the creation process but also allows for experimentation and customization of the mandalas according to the creator's preferences.

Keywords: Mathematics, Mandalas, Art, R Language.

Introdução

A geometria e os padrões matemáticos encontrados em obras de arte, como as mandalas, oferecem um terreno fértil para a exploração criativa e analítica. Uma mandala é um padrão geométrico que exibe simetria radial e auto-similaridade. Matematicamente, pode ser descrita como um conjunto de pontos no plano que é invariante sob um grupo de rotações e reflexões em torno de um ponto central. Neste contexto, este trabalho surge da

¹ Academia da Força Aérea (AFA) – luciane@id.uff.br

² Academia da Força Aérea (AFA) – jpmdosantos@yahoo.com.br



motivação em investigar e revelar a beleza oculta na complexidade das interações de cores e formas relacionadas às operações matemáticas que integram o cotidiano do ensino fundamental, médio e superior, por meio da linguagem R.

A diversidade de aplicações da geometria plana inspirou a ideia de explorar a criação de mandalas com foco no desenvolvimento matemático. Esta vasta e rica área do conhecimento, repleta de conceitos e técnicas com aplicações práticas e estéticas notáveis, suscitou o interesse em utilizar as homotetias, uma das transformações geométricas destacadas nesse campo, como base para a elaboração das mandalas. Referências diversas sobre o assunto são (PAPA NETO, 2017), (FARIA; MALTEMPI, 2012), (PRASOLOV, 2024), (FERROL *et al.*, 2017), (MONGELLI, 2008), (ALVES; COSTA; GOMES, 2020), (SRIRAMAN, 2018).

Inicialmente, as curvas paramétricas em conjunto com as rotações, translações e homotetias foram exploradas utilizando os códigos computacionais desenvolvidos em linguagem R. Experimentos preliminares resultaram em uma variedade de mandalas, caracterizadas por um complexo entrelaçamento de cores e formas geométricas. Os resultados demonstraram a utilização de princípios matemáticos e a ampla gama de opções criativas disponíveis. A introdução de elementos aleatórios na escolha das cores trouxe uma camada adicional de variação às possibilidades existentes. Os resultados obtidos seguiram a metodologia descrita em Alcoforado *et al.* (2023), onde os autores compartilharam suas descobertas e percepções sobre o tema.

A combinação única de elementos matemáticos e expressão artística presentes nas mandalas oferece um desafio estimulante e inspirador. A partir de conceitos como curvas paramétricas, homotetias e isometrias, apresenta-se um conjunto de construções representativo tanto da diversidade de resultados possíveis com diferentes técnicas de construção e coloração, quanto da precisão e profundidade da matemática subjacente. Este conjunto é utilizado para estudos, análises e exposições artísticas, oferecendo uma variedade de exemplos para examinar e apreciar.

Por fim, a abordagem inovadora no processo de coloração das mandalas (SANTOS; ALCOFORADO, 2023), juntamente com o uso da linguagem de programação R para produção e personalização, oferece uma nova perspectiva sobre a criação artística computacional.

Objetivo

O propósito deste trabalho é realizar uma exposição artística das mandalas, destacando a interseção entre arte, matemática e computação, despertando a curiosidade



dos observadores sobre os princípios matemáticos e computacionais subjacentes de sua criação.

Material e Método

A metodologia empregada neste trabalho envolveu várias etapas para a criação e seleção das mandalas a serem incluídas na exposição.

Seleção de Curvas Paramétricas: Foram escolhidas 6 curvas paramétricas como base para a criação das mandalas: a circunferência, a elipse, o cardioide, a lemniscata de Gerono, o deltóide e o astróide. As respectivas expressões paramétricas e discussões adicionais sobre estas curvas podem ser encontradas em (O'CONNOR; ROBERTSON, 2022).

Uso da Linguagem R e Ambiente R/Posit: A linguagem de programação R (R CORE TEAM, 2022), juntamente com o ambiente *RStudioPosit* (POSIT TEAM, 2024), foram utilizados para a implementação dos códigos computacionais necessários para gerar as mandalas. Isso incluiu a instalação e utilização de pacotes específicos como *ggplot2* (WICKHAM, 2023) e *dplyr* (WICKHAM; FRANÇOIS; HENRY; MÜLLER; VAUGHAN, 2023). Pormenores da utilização dos códigos podem ser encontrados em (ALCOFORADO; SANTOS; LIMA; FIRMIANO; LINARES, 2023).

Aplicação de Conceitos de Geometria: Foram aplicados conceitos de geometria, como isometrias e homotetias, para obter as composições das curvas paramétricas e criar as mandalas.

Criação das Figuras com o pacote *ggplot2* e seleção de cores: Foi utilizado o pacote *ggplot2* em conjunto com a lógica de programação, explorando-se paletas de cores disponíveis no R, com uso de métodos sequenciais e aleatórios de coloração, o que resulta em uma variedade de combinações visuais para as mandalas. Tais métodos de coloração estão apresentados em Santos e Alcoforado (2023) e Santos (2024).

Seleção dos Desenhos para Exposição: Para selecionar as mandalas a serem incluídas na exposição, uma amostra voluntária de professores e cadetes da Academia da Força Aérea (AFA) foi selecionada. Esses participantes realizaram uma votação para determinar os desenhos mais interessantes, fornecendo uma nota entre 0 e 10.

Impressão e Acondicionamento: Os desenhos mais votados foram então impressos em papel fotográfico de alta qualidade e acondicionados em molduras de tamanho A4 para compor a exposição. Esse processo garantiu uma apresentação visualmente atraente e profissional dos trabalhos selecionados.

Resultados e Discussão

As figuras selecionadas para a exposição incluem formas geométricas fundamentais como a circunferência, elipse, cardióide, lemniscata de Geronno, deltóide e astróide coloridas com diferentes métodos. Os códigos em R que geraram as figuras encontram-se no anexo. Além disso, foi produzido um [aplicativo shiny](#) para facilitar o acesso remoto à exposição.

A Figura 1 apresenta o mesmo método sequencial de coloração tendo como base a Lemniscata de Geronno (a,b,c) e a elipse (d, e, f).

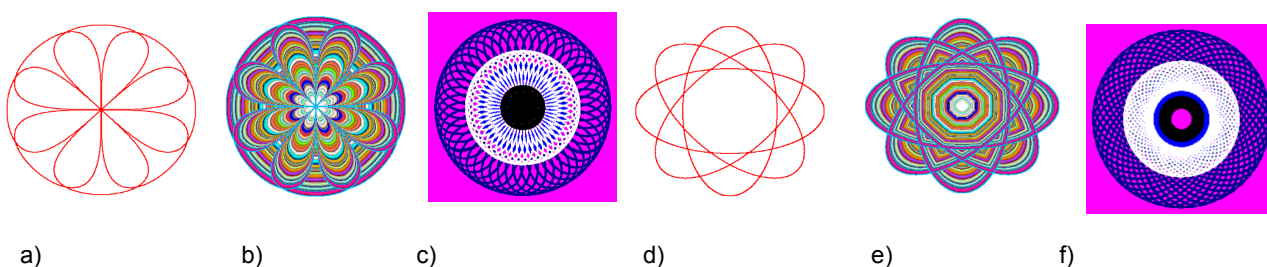


Figura 1 – Composições utilizando a mesma paleta de cores pelo método sequencial.

Fonte: Autores, 2024.

A Figura 2 apresenta composições com o método de coloração aleatório, ou seja, uma paleta de cores é escolhida e, em seguida, a escolha das cores para a composição é determinada com base em amostra aleatória com reposição e semente (`set.seed()`), tendo como base a Lemniscata de Geronno (a,b,c) e a elipse (d, e, f)

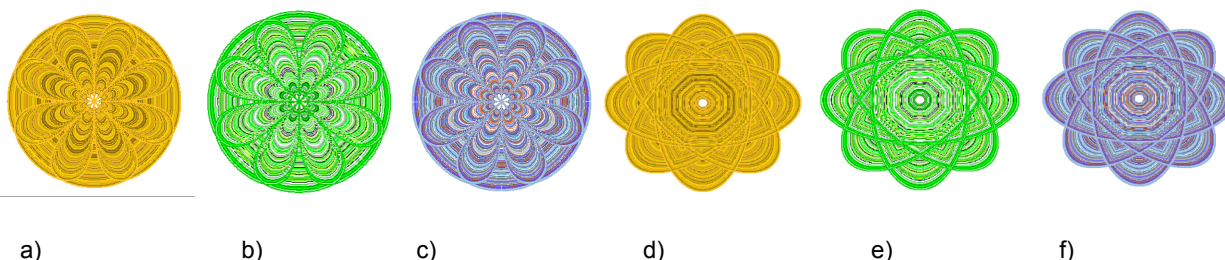
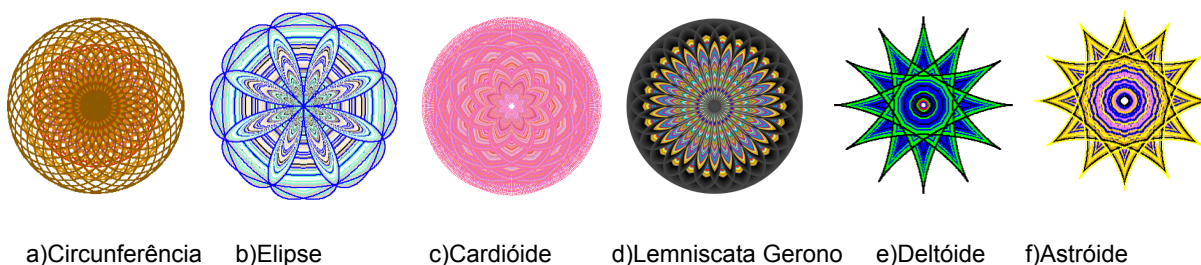


Figura 2 – Composições com diferentes paletas de cores com coloração sequencial e aleatória.

Fonte: Autores, 2024.

A Figura 3 mostra composições de cada uma das formas geométricas mencionadas, utilizando cores diversas.



a)Circunferência b)Elipse c)Cardióide d)Lemniscata Geronno e)Deltóide f)Astróide

Figura 3 – Composições com diferentes formas e paletas de cores.

Fonte: Autores, 2024.



Todas as possibilidades apresentadas são devido à utilização única e exclusiva de transformações geométricas de rotações, homotetias e translações. Para efeitos educacionais, é possível introduzir a lógica de programação a partir dos conceitos matemáticos envolvidos na construção das mandalas. Do ponto de vista artístico é possível explorar os conceitos de estética e harmonia das cores.

Conclusão

Neste trabalho, apresentamos os principais resultados provenientes da abordagem para a geração de Mandalas com base em elementos matemáticos, proporcionados pela utilização combinada da linguagem R. O objetivo principal foi realizar uma exposição artística das mandalas geradas a partir da interseção entre arte, matemática e computação.

A seleção das peças para a exposição foi um processo colaborativo, baseado nas opiniões de uma amostra representativa de cadetes e professores da Academia da Força Aérea (AFA). Este método de curadoria enfatiza a importância da interação comunitária no processo criativo.

Uma versão online da exposição foi desenvolvida como um aplicativo Shiny para facilitar o acesso dos participantes.

As mandalas exibidas são mais do que meras representações artísticas; elas são um exemplo de educação interdisciplinar unindo arte, lógica, matemática e computação. Através desta exposição, esperamos ter despertado a curiosidade intelectual, incentivando uma exploração mais profunda da arte computacional e suas possibilidades ilimitadas.

Referências

- ALCOFORADO, L.F., SANTOS, J.P.M., LIMA, M.V.A., JESUS, A.F., LINARES, J.L. **Mandalas, Curvas Clássicas e Visualização com R**. Universidade de São Paulo. Faculdade de Zootecnia e Engenharia de Alimentos, 2023. DOI: <https://doi.org/10.11606/9786587023335>. Disponível em: www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/book/1017. Acesso em 14 abril. 2024.
- ALVES, D. S. P., COSTA, L. M. G. C., GOMES, J. D. S. J. Visualização em Ambientes Dinâmicos como Facilitador no Ensino de Simetrias e Pavimentações. **Revista Brasileira de Expressão Gráfica**: Vol. 8, No. 1, 2020.
- FERRÉOL, R.; BOUREAU, S.; ESCULIER, A. **2d curves**. 2017. Disponível em: <https://mathcurve.com/courbes2d.gb/courbes2d.shtml>. Acesso em: 29 abr. 2022.
- FARIA, R.; MALTEMPI, M. Padrões fractais: conectando matemática e arte. **EccoS – Rev. Cient.** v. 27, p. 33-53, 2012.
- MONGELLI, Magda Cristina Junqueira Godinho. **Desenho Geométrico**, Ed. UFMS, Campo Grande. 2008.
- O'CONNOR, J.; ROBERTSON, E. Índice de curvas famosas. MacTutor, 2022. Disponível em: <https://mathshistory.st.andrews.ac.uk/Curves/>. Acesso em: 26 out. 2024.
- PAPA NETO, Angelo. **Geometria plana e construções geométricas**. Fortaleza: UAB/IFCE, 2017. 226 p. ISBN 978-85-475-0059-7.
- POSIT TEAM. **RStudio: Integrated Development Environment for R**. Posit. Software, PBC, Boston, MA. 2024. URL <http://www.posit.co/>. Acesso em 10/05/2024.
- PRASOLOV, Viktor. **Problems in Plane and Solid Geometry**. Vol. 1: Plane Geometry. Traduzido e editado por Dimitry Leites. Disponível em: <https://imogeometry.blogspot.com/p/geometry-books-pdf.html>{Geometry Problems from IMOs. Acesso em 14 abril 2024.
- R CORE TEAM. **R: A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria. 2022. URL <https://www.R-project.org/>. Acesso em 10/05/2024.



SANTOS, J.P.M.; ALCOFORADO, L.F. **Colorindo mandalas com R: explorando cores e gradientes em curvas planas**. In: Maria José Ginzo Villamayor. Programa de Resumos: X Xornada de usuarios de R em Galicia. Cidade: Santiago de Compostela. Asociación de usuarios de software libre da Terra de Melide, p.47-50, 2023. ISBN: 978-84-09-55129-3.

SANTOS, J.P.M. Curvas e Cores em R: Movimentos Rígidos no Plano. In: ALCOFORADO, L. F., SANTOS, J.P.M, LEVY, A., LONGO, O.C., LINARES, J. L. **Aplicações em R : encurtando distâncias nas ciências**. Universidade de São Paulo. Faculdade de Zootecnia e Engenharia de Alimentos, 2024. Disponível em: www.livrosabertos.abcd.usp.br/portaldelivrosUSP/catalog/book/1249. Acesso em 15 abril 2024.

SRIRAMAN, B. (Ed.). **Handbook of the Mathematics of the Arts and Sciences**. Cham: Springer, 2018. DOI: <https://doi.org/10.1007/978-3-319-70658-0>. Acesso em 10/05/2024.

WICKHAM, Hadley. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. Disponível em: <https://ggplot2.tidyverse.org>. Acesso em 10/05/2024.

WICKHAM, Hadley; FRANÇOIS, Romain; HENRY, Lionel; MÜLLER, Kirill; VAUGHAN, Davis. dplyr: A Grammar of Data Manipulation. 2023. R package version 1.1.2. Disponível em: <https://CRAN.R-project.org/package=dplyr>. Acesso em 10/05/2024.



Anexo – Código R

```
#Figura1a e 1b
require(ggplot2)
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt,yt)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=colors()[i],size=size) }

#Figura 1c
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta)
z=rep(0,n); rotacao=c(seq(0,2*pi,0.125))
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
}

dt=tibble::tibble(xt,yt)
contracao = 0.25;xt2=xt*contracao; yt2=yt*contracao; dt2=tibble::tibble(xt2,yt2)
contracao = 0.5;xt3=xt*contracao;yt3=yt*contracao; dt3=tibble::tibble(xt3,yt3)
contracao = 0.65;xt4=xt*contracao;yt4=yt*contracao; dt4=tibble::tibble(xt4,yt4)
size=0.25
p= ggplot()+ coord_fixed()+ theme_void()
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="blue4",size=size)
p = p+geom_point(data=dt3, aes(x=xt3, y=yt3), color="blue",size=size)
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color="white",size=size)
p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color="black",size=size)
p=p +theme(panel.background = element_rect(fill = "magenta") )
p

#Figura1d e 1e
require(ggplot2)
n=1000;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=2*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt,yt)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=colors()[i],size=size) }
```



```
#Figura1e
require(ggplot2)
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=2*cos(theta)
z=rep(0,n); rotacao=c(seq(0,2*pi,0.125))
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))}

dt=tibble::tibble(xt,yt)
contracao = 0.25;xt2=xt*contracao; yt2=yt*contracao; dt2=tibble::tibble(xt2,yt2)
contracao = 0.5;xt3=xt*contracao;yt3=yt*contracao; dt3=tibble::tibble(xt3,yt3)
contracao = 0.65;xt4=xt*contracao;yt4=yt*contracao; dt4=tibble::tibble(xt4,yt4)
size=0.25
p= ggplot()+ coord_fixed()+ theme_void()
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="blue4",size=size)
p = p+geom_point(data=dt3, aes(x=xt3, y=yt3), color="blue",size=size)
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color="white",size=size)
p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color="black",size=size)
p=p +theme(panel.background = element_rect(fill = "magenta") )
p

#Método Aleatório de coloração
#Figura 2a
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt,yt)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=sample(colors()[142:151],length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 2b
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt,yt)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=sample(colors()[251:261],length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))
```




```
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 2c
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
  p= ggplot()+ coord_fixed()+ theme_void()
  dt=tibble::tibble(xt,yt)
  step=0.0075; contracao = seq(.1,1,by=step);size=0.25
  set.seed(10)
  cores=sample(colors()[580:605],length(contracao),replace=T)
  MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="seashell3",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 2d
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=2*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
  p= ggplot()+ coord_fixed()+ theme_void()
  dt=tibble::tibble(xt,yt)
  step=0.0075; contracao = seq(.1,1,by=step);size=0.25
  set.seed(10)
  cores=sample(colors()[142:151],length(contracao),replace=T)
  MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 2e
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=2*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
```



```
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt,yt)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=sample(colors())[251:261],length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 2f
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=2*cos(theta)
z=rep(0,n); rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt=x; yt=y
for(i in 1:length(rotacao)){
  xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
  yt=c(yt,y[1:n]*sin(rotacao[i])+x[1:n]*cos(rotacao[i]))
}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt,yt)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=sample(colors())[580:605],length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="seashell3",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt*contracao[i])
  yt2=c(yt*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 3a
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=cos(theta)
xt=c(x+1); yt=c(y)
n1=16; rotacao=pi/n1*1:(2*n1)
xt1=xt; yt1=yt

for(i in 1:length(rotacao)){
  xt1=c(xt1,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
  yt1=c(yt1,yt[1:n]*sin(rotacao[i])+xt[1:n]*cos(rotacao[i]))
}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt1,yt1)
step=0.45; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=sample(colors())[485:500],length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt1, y=yt1), color="seashell3",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt1*contracao[i])
  yt2=c(yt1*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p
```



```
#Figura 3b
n=500;theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=2*cos(theta)
xt=c(x+1); yt=c(y)
n1=4; rotacao=pi/n1*1:(2*n1)
xt1=xt; yt1=yt

for(i in 1:length(rotacao)){
  xt1=c(xt1,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
  yt1=c(yt1,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt1,yt1)
step=0.015; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=sample(colors()[1:27],length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt1, y=yt1), color="oldlace",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt1*contracao[i])
  yt2=c(yt1*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 3c
n=500;theta=seq(0,2*pi, length.out = n);raio=1
x=c(2*raio*cos(theta)-raio*cos(2*theta))
y=c(2*raio*sin(theta)-raio*sin(2*theta))
xt=c(x+0); yt=c(y)
n1=4; rotacao=pi/n1*1:(2*n1)
xt1=xt; yt1=yt

for(i in 1:length(rotacao)){
  xt1=c(xt1,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
  yt1=c(yt1,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt1,yt1)
step=0.015; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=c("indianred", "lightcoral", "salmon", "lightsalmon", "plum", "orchid", "pink", "violet", "hotpink")
cores=sample(cores,length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt1, y=yt1), color="black",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt1*contracao[i])
  yt2=c(yt1*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p

#Figura 3d
n=500;theta=seq(0,pi, length.out = n);raio=1
x=sin(theta); y=sin(theta)*cos(theta) #(lemniscata)
xt=c(x+0); yt=c(y)
n1=13; rotacao=pi/n1*1:(2*n1)
xt1=xt; yt1=yt

for(i in 1:length(rotacao)){
  xt1=c(xt1,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
  yt1=c(yt1,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))
```



```
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt1,yt1)
step=0.005; contracao = seq(.1,1,by=step);size=0.25
p=p+ geom_point(data=dt, aes(x=xt1, y=yt1), color="black",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt1*contracao[i])
  yt2=c(yt1*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=colors()[i],size=size) }
p

#Figura 3e
n=500;theta=seq(0,pi, length.out = n);raio=1
x=2*cos(theta)+cos(2*theta); y=2*sin(theta)-sin(2*theta) #deltoide
xt=c(x+0); yt=c(y)
n1=6; rotacao=pi/n1*1:(2*n1)
xt1=xt; yt1=yt

for(i in 1:length(rotacao)){
  xt1=c(xt1,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
  yt1=c(yt1,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt1,yt1)
step=0.05; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=c("red","green", "blue", "black")
cores=sample(cores,length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt1, y=yt1), color="black",size=size)

for(i in 1:length(contracao)){
  xt2=c(xt1*contracao[i])
  yt2=c(yt1*contracao[i])
  dt2=tibble::tibble(xt2,yt2)
  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p
```

```
#Figura 3f
n=500;theta=seq(0,pi, length.out = n);raio=1
x=3*cos(theta)^3; y=3*sin(theta)^3 #astroide
xt=c(x+0); yt=c(y)
n1=6; rotacao=pi/n1*1:(2*n1)
xt1=xt; yt1=yt

for(i in 1:length(rotacao)){
  xt1=c(xt1,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
  yt1=c(yt1,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))}
p= ggplot()+ coord_fixed()+ theme_void()
dt=tibble::tibble(xt1,yt1)
step=0.05; contracao = seq(.1,1,by=step);size=0.25
set.seed(10)
cores=c("orange","yellow","blue", "gray", "goldenrod","gold", "blue3","violet","black")
cores=sample(cores,length(contracao),replace=T)
MinhasCores=rep(cores,floor(length(contracao)/length(cores)))

p=p+ geom_point(data=dt, aes(x=xt1, y=yt1), color="black",size=size)

for(i in 1:length(contracao)){
```



REVISTA DO SEMINÁRIO INTERNACIONAL DE ESTATÍSTICA COM R | ISSN: 2526-7299

VOL 5, Nº 2, JULHO DE 2024

VIII SEMINÁRIO INTERNACIONAL DE ESTATÍSTICA COM R - AI IN DATA SCIENCE

```
xt2=c(xt1*contracao[i])
yt2=c(yt1*contracao[i])
dt2=tibble::tibble(xt2,yt2)
p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size) }
p
```