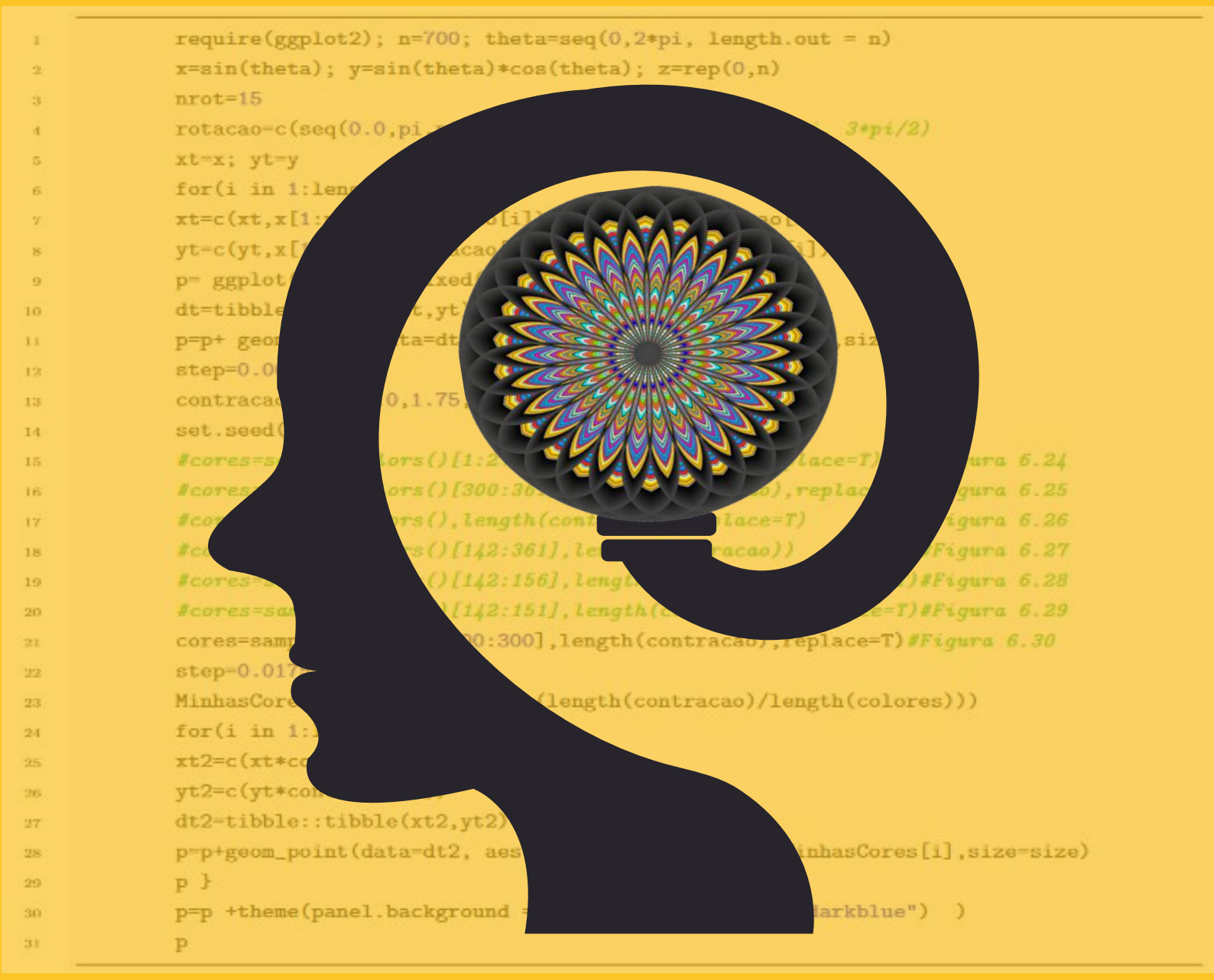
MANDALAS

CURVAS CLÁSSICAS E VIZUALIZAÇÃO COM R



LUCIANE ALCOFORADO
JOÃO PAULO M SANTOS
ALESSANDRO FIRMIANO
MARCUS VINICIUS LIMA
JUAN LÓPEZ LINARES

LUCIANE FERREIRA ALCOFORADO
JOÃO PAULO MARTINS DOS SANTOS
MARCUS VINÍCIUS DE ARAUJO LIMA
ALESSANDRO FIRMIANO DE JESUS
JUAN LÓPEZ LINARES

Mandalas, curvas clássicas e visualização com R

DOI: 10.11606/9786587023335

Pirassununga - SP
FACULDADE DE ZOOTECNIA E ENGENHARIA DE ALIMENTOS (FZEA)
UNIVERSIDADE DE SÃO PAULO (USP)
2023

UNIVERSIDADE DE SÃO PAULO

Reitor: Prof. Dr. Carlos Gilberto Carlotti Junior

Vice-Reitora: Profa. Dra. Maria Arminda do Nascimento Arruda

FACULDADE DE ZOOTECNIA E ENGENHARIA DE ALIMENTOS

Avenida Duque de Caxias Norte, 225 - Pirassununga, SP

CEP 13.635-900

http://www.fzea.usp.br

Diretor: Prof. Dr. Carlos Eduardo Ambrósio

Vice-Diretor: Prof. Dr. Carlos Augusto Fernandes de Oliveira

Dados Internacionais de Catalogação na Publicação

Serviço de Biblioteca e Informação da Faculdade de Zootecnia e Engenharia de Alimentos da Universidade de São Paulo

A354m

Alcoforado, Luciane Ferreira
Mandalas, curvas clássicas e visualização com R. / Luciane Ferreira Alcoforado, João Paulo Martins dos Santos, Marcus Vinícius de Araújo Lima, Alessandro Firmiano de Jesus, Juan López Linares. -- Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos da Universidade de São Paulo, 2023.
105 p.

ISBN 978-65-87023-33-5 (e-book)
DOI: 10.11606/9786587023335

1. Geometria. 2. Mandalas. 3. Software R. 4. Isometrias. 5. Homotetias. I. Santos, João Paulo Martins dos. II. Lima, Marcus Vinícius de Araújo. III. Jesus, Alessandro Firmiano de. IV. Linares, Juan López. V. Título.

Ficha catalográfica elaborada por Girlei Aparecido de Lima, CRB-8/7113

Dedicamos este livro a nossas famílias.

AGRADECIMENTOS

Agradecemos a todos que incentivaram de forma direta ou indireta na produção deste e-book. Uma obra para ser concretizada depende da colaboração de inúmeras pessoas. As leituras, opiniões e críticas construtivas muito contribuíram para desenvolver, principalmente, a técnica de coloração das figuras. Agradecemos a nossas famílias pelo incentivo e compreensão. Finalmente agradecemos às Instituições apoiadoras, Academia da Força Aérea e Universidade de São Paulo, que possibilitaram a parceria entre os autores.

AUTORES

Dra. LUCIANE FERREIRA ALCOFORADO: https://orcid.org/0000-0002-9504-8087. Possui graduação em Licenciatura em Matemática pela Universidade Federal de Santa Maria (1994), mestre em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (1998) e Doutora em Engenharia Civil pela Universidade Federal Fluminense (2009). É professora na Academia da Força Aérea em Pirassununga/SP e professora colaboradora no Programa de Pós-Graduação em Engenharia Civil da UFF. Tem experiência na linguagem R, análise e visualização de dados e em problemas de otimização. Sua atuação na área de Matemática e Estatística destaca-se nos seguintes temas: regressão linear, regressão logística, testes de hipóteses, análise multivariada, métodos de apoio à tomada de decisão como Simplex, AHP entre outros. É líder do grupo de Pesquisa Estatística com R, coordenadora do SER - Seminário Internacional de Estatística com R, autora de diversos livros sobre a linguagem R. Textos completos e gratuitos das publicações da autora podem ser encontrados aqui.

Dr. JOÃO PAULO MARTINS DOS SANTOS: https://orcid.org/0000-0002-0957-7119. Possui graduação em Licenciatura em Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (2006), mestre em Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (2009) e Doutor em Ciências pela Escola de Engenharia de São Carlos - EESC-USP. É professor na Academia da Força Aérea em Pirassununga/SP. Tem experiência na área de Sistemas Dinâmicos não lineares e não ideais com pesquisa desenvolvida em métodos de perturbação. Tem experiência na área de Matemática e interesse nos seguintes temas: método numéricos para solução de equações diferenciais ordinárias e parciais, estimador de erro do tipo residual para a equação do transporte de poluentes, linguagem Python de programação, Computação Científica em Python e métodos numéricos para solução de sistemas lineares. Textos completos e gratuitos das publicações do autor podem ser encontrados aqui.

Dr. MARCUS VINÍCIUS DE ARAÚJO LIMA: https://orcid.org/0000-0003-3675-566X Possui graduação em Matemática pela Universidade Federal de São Carlos (1993), mestrado em Matemática pela Universidade Federal de São Carlos (1997) e doutorado em Matemática pela Universidade Federal de São Carlos (2001). Atualmente é professor associado 2 da Academia da Força Aérea (AFA), em Pirassununga. Tem experiência na área de Matemática, com ênfase em Física Matemática tendo atuado principalmente nos seguintes temas: espectro de operadores de Schröedinger discretos unidimensionais com potenciais de substituição, sequências de substituição primitivas, sequências de substituição não primitivas, espectro singular contínuo. O currículo Lattes pode ser acessado em http://lattes.cnpq.br/8543074981120879.

Dr. ALESSANDRO FIRMIANO DE JESUS: https://orcid.org/0000-0002-7073-2261. Possui graduação em Matemática pela Universidade de São Paulo (1994), mestrado em Matemática pela Universidade Federal de São Carlos (1997), doutorado em Engenharia Hidráulica e Saneamento pela Universidade de São Paulo (2010) e pós doutorado pela Fachhochschule

Köln - Cologne University of Applied Sciences (2014). Atualmente é regime jurídico único da Academia da Força Aérea AFA exercendo a função de Prof. Associado IV. Tem experiência na área de Matemática, com ênfase em Modelagem Computacional e Análise Numérica, atuando principalmente nos seguintes temas: Modelos de advecção-difusão-reação, Método dos Elementos Finitos, Transporte de Contaminantes em Águas Subterrâneas e Geocomputação. Coordenador da Área de Ciências Exatas no Curso de Formação de Oficiais da Aeronáutica na Academia da Força Aérea (AFA). Textos completos e gratuitos das publicações do autor podem ser encontrados aqui.

Dr. JUAN LÓPEZ LINARES: https://orcid.org/0000-0002-8059-0631.
Professor Associado do Departamento de Ciências Básicas (ZAB) da Faculdade de Zootecnia e Engenharia de Alimentos (FZEA) da Universidade de São Paulo (USP). Atualmente ministra as disciplinas de Cálculo II e IV para estudantes de engenharias e os cursos de "Treinamento Olímpico em Matemática para estudantes do Ensino Fundamental e Médio" e "Geometria olímpica com GeoGebra" para professores. Desenvolve projetos de pesquisa nas áreas de ensino de Cálculo e na resolução de problemas de Olimpíadas. Graduação e Mestrado em Física na Universidade da Havana, Cuba, em 1994 e 1996, respetivamente. Curso de Diploma da Matéria Condensada no Centro Internacional de Física Teórica Abdus Salam, em Trieste, na Itália em 1997-1998. Estágio no Instituto de Espectroscopia Molecular (CNR), Bolonha, Itália em 1998-1999. Doutor em Física pela Universidade Federal de São Carlos (UFSCar) em 1999-2001. Pós-doutorado de 4 anos (2002-2005) na Universidade Estadual de Campinas (Unicamp). Mestre Profissional em Matemática em Rede Nacional (PROFMAT) pela UFSCar em 2019 e Livre Docente na área de Ensino de Matemática Olímpica na FZEA USP em 2022. Textos completos e gratuitos das publicações do autor podem ser encontrados aqui.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Título

Mandalas, curvas clássicas e visualização com R

Prefácio

Este material é fruto da palestra de mesmo título "MANDALAS, CURVAS CLÁSSICAS E VISUALIZAÇÃO COM R", apresentada no VI Seminário Internacional de Estatística com R (SER), ocorrido em 2022 em modo online. Com foco na multidisciplinaridade, o evento SER possui foco na utilização da linguagem R em diferentes contextos, tais como Estatística, Ciências Sociais, Economia e Finanças, Computação, Matemática, Biologia, entre outras. Nesse contexto da diversidade, o material expande a utilização da linguagem R e mostra que sua capacidade computacional pode ser aproveitada em outras áreas e utilizadas até em contextos do Ensino Fundamental e Médio. O material traz uma apresentação detalhada para a construção de Mandalas por meio do software R. Isso possibilita englobar a diversidade de utilização da linguagem R em um único elemento, pois engloba as construções gráficas, transformações matemáticas e a programação computacional. As construções, inicialmente, são focadas na apresentação de alguns detalhes matemáticos e detalhes históricos; em seguida estão mostrados os códigos completos e diretos para a construção das mandalas utilizando variáveis cartesianas; as coordenadas polares são utilizadas logo em seguida. Em cada caso, os códigos completos são apresentados para facilitar a reprodução com consequente análise e entendimento. A discussão é organizada em sete capítulos: introdução; R, RStudio/Posit e pacotes tible e ggplot2; conceitos de Matemática e Programação; Transformações; Mandalas; Cores em R e Referências. O foco são as construções geométricas detalhadas com a apresentações passo a passo e o compartilhamento dos códigos para que o leitor possa acompanhar e explorar o processo descrito.

Palavras-chave: Geometria, Mandalas, Software R, Isometrias, Homotetias

Lista de Figuras

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Sumário

Capítulo 1

Introdução

O livro faz parte de um projeto de desenvolvimento e construção de mandalas usando curvas clássicas e programação computacional utilizando linguagem R. O texto foi elaborado com base nas anotações iniciais da palestra intitulada **Mandalas, curvas clássicas e visualização com R**, apresentada por dois dos autores no *VI International Seminar on Statistics with R*, em 2022, online, disponível em https://ser.uff.br/. A boa repercussão dessa palestra motivou os autores a expandir e detalhar o processo de construção de mandalas usando o R, que resultou neste *e*-book.

A proposta evoluiu do interesse dos autores em aplicar as possibilidades do uso da linguagem R, com o objetivo de estimular a aprendizagem e o uso dessa linguagem, particularmente à partir da exploração da relação entre matemática e arte presente, por exemplo, nas mandalas.

O material não é uma proposta de manual, apesar de quase todas as figuras virem acompanhadas dos códigos completos que as geram. Por um lado, isso acarreta uma certa redundância para o leitor familiarizado com a linguagem R, por outro, deixa o texto mais completo para o leitor iniciante na linguagem, permitindo a utilização direta dos códigos para a reprodução das figuras e evitando a busca por fragmentos de códigos espalhados pela obra.

Os códigos e, consequentemente, as figuras são apresentadas em uma sequência crescente de complexidade, iniciando com figuras simples como circunferência e elipse, passando para outras curvas como cardioide e lemniscata, inserindo rotações e homotetias para a geração das mandalas e, por fim, explorando a inserção de cores à construção das figuras. Para além da criação das figuras, a preocupação foi a de se estabelecer um método que pudesse ser reproduzido para diversas curvas, explorando desde a concepção de uma base de dados que surge da necessidade de informar o conjunto de pontos que formarão a figura à partir da escolha de uma curva, passando pelas equações matemáticas que descrevem as curvas, as transformações lineares que produzem os efeitos finais, chegando à descrição explicativa do código em R para a obtenção do produto final.

O livro é organizado em sete capítulos. O Capítulo 2 discorre sobre o R, RStudio/RPosit e

Pacotes *tibble* e *ggplot2*, mostrando como obter os programas, como instalar e como utilizar para o propósito deste livro. No Capítulo 3 são apresentadas as curvas que serão usadas na geração das mandalas, além dos códigos que as geram, que são essenciais para o desenvolvimento dos capítulos posteriores. O Capítulo 4 apresenta as transformações de rotação e homotetia no plano cartesiano, bem como as aplicações às curvas planas apresentadas anteriormente, com respectivos códigos computacionais. O Capítulo 5 discute a geração das Mandalas por meio de um processo construtivo que tem início na escolha de uma ou mais curvas planas, combina transformações de rotações e homotetias as transformações geométricas e, por fim, a respectiva implementação computacional com apresentação detalhada dos códigos em linguagem R. No Capítulo 6, denominado, Cores em R, as cores são inseridas no processo de construção para geração de figuras ricas em detalhes e com belo apelo visual.

As Referências Bibliográficas trazem a fundamentação para tornar essa obra possível, com destaque para as contribuições significativas das referências [22], [24], [35], [1] e [10].

1.1 Arte e Matemática

Muito do que é desenvolvido neste livro remete ao processo criativo e a aplicação de conceitos matemáticos da lógica, da geometria e da programação computacional.

A Matemática é uma ciência natural que surgiu da relação entre o ser humano e a natureza. Presente na vida cotidiana, relaciona-se como ferramenta e linguagem com diversas áreas como engenharias, física, química, biologia, entre outras. Em particular, nas artes visuais é comum encontrar figuras geradas pela aplicação sistemática de padrões geométricos, simetrias e semelhanças. Sistematizar essa combinação de formas e propriedades geométricas por meio de algoritmos, é uma forma de produzir figuras com apelo artístico.

De acordo com Faria e Maltempi [4], a Matemática e a Arte possuem o potencial de revelar estruturas e padrões que nos auxiliam a compreender o mundo e a desenvolver a imaginação de forma estruturada. Devido à relação com a visualização, o campo da Matemática que está mais relacionada com a Arte é a Geometria. Neste contexto, perspectiva, proporção e simetria, são alguns dos exemplos fundamentais dessa relação.

A arte é conhecimento e traz conhecimento, faz pensar e agir, e como coloca Puccetti e Souza (2011, p. 2511), [23], "o processo de criação artística é, em si, um processo de conhecimento" a partir do momento em que traz relações que conectam o sujeito que compreende, relaciona, transforma e cria, percebendo-se como um ser criativo, capaz de transpor os seus limites.

Nas palavras de Martins, Picosque e Guerra (2010, p.127), [11]:

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

> Há um instante mágico na vida em que, nem mesmo sabendo por que, ficamos envolvidos num jogo. Num jogo de aprender e ensinar. Fazemos parcerias. Não só com os outros, mas também parcerias internas nos propondo desafios. Porém, só ficamos nesse estado de total cumplicidade com o saber se este tem sentido para nós. Caso contrário, somos apenas espectadores do saber do outro.

Durante o desenvolvimento deste e-book, nos deparamos com termo $A/R/TOGRAFIA$ cunhado em 2004 por Rita Irwin [7]. É uma linha de investigação inserida na metodologia denominada de Pesquisa Educacional Baseada em Arte (PEBA) e descrito por [14] como:

> A a/r/tografia é uma metodologia de pesquisa derivada da "investigação baseada nas artes", ou seja, é uma prática da "investigação baseada nas artes", igualmente de perspectiva narrativa que parte do acrônimo a/r/t "a"de artist, "r"de researcher e "t"de teacher (em língua portuguesa, respectivamente, artista, investigador e professor). Já o termo graphy, na sua etimologia grega (graphein), significa "escrever, representar graficamente". A a/r/tografia seria, então, um tipo de pesquisa realizada/produzida por um pesquisador que exerce também função de professor e artista concomitantemente (Entendendo que o artista poderá ser músico, poeta, dançarino, ator, performer, escultor, pintor, gravador, etc.).

De acordo com [13], o que diferencia a a/r/tografia dos outros métodos da PEBA é a produção artística realizada durante o percurso da investigação. Ao passo que, em outras condições de PEBA, as artes são utilizadas apenas como dados para análise, no método a/r/tográfico, o processo de criação e os movimentos que surgem dele também são relevantes para a pesquisa. Neste sentido, este livro produz uma pequena contribuição para pesquisadores que atuam nesta linha de investigação e pretendam fazer uso de uma linguagem computacional para o processo de criação.

Um outro aspecto interessante do aprendizado pode ser caracterizado pela interação entre Ciência, Tecnologia, Engenharia e Matemática (STEM (em Inglês *Science, Technology, Engineering, and Math*)) ou de forma mais ampla por Ciência, Tecnologia, Engenharia, Arte, e Matemática (STEAM (em Inglês *Science, Technology, Engineering, Art, and Math (STEAM)*)) [9]. Segundo estes autores existem múltiplas perspectivas sobre o significado do termo STEM, o que dificulta a classificação de quais atividades escolares podem ser classificadas em STEM ou STEAM. A classificação das disciplinas individuais ou das interações entre as disciplinas varia de acordo com o posicionamento de cada autor. Uma lista de autores e pontos de vista é fornecida em [9].

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

1.2 Mandalas

A proposta deste e-book é mostrar como podemos utilizar diversos recursos de programação, visualização e formas parametrizadas de figuras geométricas para produzir mandalas. A mandala, de acordo com a professora de História Juliana Bezerra [3] é uma circunferência que contém em seu interior desenhos de formas geométricas, figuras humanas e cores variadas. De acordo com o dicionário *Oxford Languages*, a palavra mandala é classificada como substantivo masculino, mas observamos seu uso como substantivo feminino. De acordo com o dicionário, a palavra está classificado como Filosofia e Religião, e seu significado é um diagrama de formas geométricas concêntricas.

Levando em conta as discussões da Seção 1.1, as mandalas podem ser um recurso didático utilizado por professores de Arte, História e Matemática, pois serve para ilustrar tópicos tais como as formas geométricas, cores, percepção visual, história da arte, história das religiões [3]. No contexto da programação e utilização de tecnologias de ensino, a linguagem R pode ser explorada para inserir elementos de programação computacional elementar ou avançada. Por fim, no presente contexto, a Matemática é explorada por meio das curvas clássicas com transformações rígidas e visualização gráfica no plano .

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Capítulo 2

R, RStudio e Pacotes *tibble* e *ggplot2*

A linguagem R está consolidada no meio de analistas de dados. Trata-se de uma linguagem de programação especializada em Estatística, Análise e Visualização de dados. No entanto sua aplicabilidade vai muito além da análise de dados, podendo ser utilizada para produção de relatórios, livros, apresentações e desenvolvimento de pacotes e aplicativos web, além de produções artísticas a partir de conceitos matemáticos e de programação, como é o caso do que se propõe este e-book.

Segundo [2], o software R surgiu na década de 1990 na Universidade de Auckland, fruto da iniciativa de professores de Estatística, Robert Gentleman e Ross Ihaka. A primeira versão oficial foi publicada em fevereiro de 2000. No Brasil, o primeiro espelho do *Comprehensive R Archive Network* (CRAN) do R foi iniciado pelo departamento de Estatística da Universidade Federal do Paraná (UFPR) por iniciativa do professor Paulo Justiniano que disponibilizou as primeiras apostilas na língua portuguesa.

Atualmente, de acordo com [1], o interesse hoje em aprender R está difundido em diversas profissões, sendo muito simples e rápida sua instalação, bastando para isso acessar o site do projeto R em https://cran.r-project.org/. Outra ferramenta que compõe o ambiente de programação da linguagem R é o ambiente de desenvolvimento integrado denominado de R Studio/Posit, que pode ser obtido em https://www.rstudio.com/products/rstudio/download/.

A linguagem R é composta de um grande número de pacotes que são um conjunto de funções programadas para executar diversos cálculos e procedimentos computacionais. Após instalar o R e o RStudio/Posit, um conjunto destes pacotes se tornam automaticamente disponíveis, enquanto que outros necessitam ser instalados de acordo com a necessidade do usuário. Isso se dá pois o rol de pacotes disponíveis aproxima-se de 20 mil pacotes e nem todos se fazem necessários para cada usuário. A lista de pacotes é atualizada diariamente e pode ser acessada em https://cran.r-project.org/.

Para a construção de mandalas utilizadas neste e-book utilizaremos os pacotes descritos a seguir.

2.1 *tibble*

O pacote *tibble* foi desenvolvido por [12] como uma versão moderna das tabelas de dados do pacote básico da linguagem R. Com este pacote é possível criar e manipular estruturas de uma tabela (*dataframe*). Para fazer uso do pacote é necessário sua instalação por meio do comando:

```
1    install.packages("tibble")
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *tibble*.

Uma tabela de dados pode ser entendida como um conjunto de vetores colunas, cada coluna representa uma variável e cada linha representa uma observação desta variável.

Suponha que precisamos criar a seguinte tabela de dados:

x	4	5	6
y	0	0	0

O código a seguir ilustra o uso do pacote *tibble* para criar a tabela que será denominada de "tabela":

```
1    tabela = tibble::tibble(x=c(4,5,6), y=c(0,0,0))
2    tabela
```

O detalhamento do código é feito a seguir:

1. Cria uma tabela utilizando a função *tibble* do pacote *tibble*, o primeiro *tibble* refere-se ao pacote seguido de :: e do nome da função do respectivo pacote;

2. Exibe o conteúdo de tabela.

O comando da linha 1 um cria a tabela com duas colunas, a primeira com o vetor x e a segunda com o vetor y. A linha 2 imprime o objeto tabela.

Caso a tabela contenha mais de 10 linhas, ao imprimir ou visualizar o objeto *tibble*, apenas as 10 primeiras linhas serão exibidas, caso contrário será necessário informar o número de linhas que devem ser exibidas.

Ilustramos a seguir como visualizar apenas uma coluna ou algumas colunas não consecutivas do objeto *tibble*.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

```
1  #constrói tabela de dados
2  tabela = tibble::tibble(x=c(4,5,6), y=c(0,0,0), z=c("a", "b", "c"))
3  tabela$x #visualiza apenas a coluna denominada x
4  [1] 4 5 6
```

O detalhamento do código é feito a seguir:

1. Comentário;

2. Cria uma tabela utilizando a função *tibble* contendo 3 colunas: x, y e z;

3. Exibe o conteúdo da coluna x da tabela;

4. Resultado do comando da linha 3.

```
1      tabela$z #visualiza apenas a coluna denominada z
2          [1] "a" "b" "c"
```

O detalhamento do código é feito a seguir:

1. Exibe o conteúdo da coluna z da tabela;

2. Resultado do comando da linha 1.

```
1      tabela[ , c(1,3)] #vizualiza colunas 1 e 3, ou seja x e z
2      # A tibble: 3 x 2
3                 x     z
4               <dbl> <chr>
5          1     4      a
6          2     5      b
7          3     6      c
```

O detalhamento do código é feito a seguir:

1. Exibe o conteúdo das colunas 1 e 3. Após # é comentário.

2. Exibição do resultado do comando da linha 1, inicia com informações sobre a dimensão da tabela, sendo 3 linhas e 2 colunas.

3. Exibe o nome das colunas solicitadas no comando da linha 1.

4. Exibe o tipo de variável de cada coluna, `dbl` significa *double* e *chr* significa caractere.

5-7. Exibe o valores das colunas 1 e 3 da tabela, com as linhas numeradas de 1 a 3 respectivamente.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

2.2 *ggplot2*

O pacote *ggplot2* foi desenvolvido por [35] como ferramenta poderosa para a visualização de dados. Com este pacote é possível criar diversos tipos de gráficos. Foi idealizado para que a construção fosse um processo em camadas, ou seja, a construção do pacote permite acrescentar ou modificar camadas do gráfico de forma prática.

Para fazer uso do pacote é necessário sua instalação por meio do comando:

```
1    install.packages("ggplot2")
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

Para realizar a construção do gráfico, é necessário que os dados estejam armazenados em uma tabela (*tibble*), contendo pelo menos duas colunas, uma com os valores do eixo x e outra com os valores do eixo y.

O *ggplot2* permite diversas formas para o gráfico por meio da família de argumentos do tipo "geom_". Por exemplo geom_histogram produz a forma de um histograma, geom_boxplot produz a forma de um *boxplot*, geom_point produz pontos. A lista completa pode ser vista na documentação do pacote.

A título de ilustração, o código a seguir mostra o processo de visualização de um conjunto de pontos qualquer.

```
1    tabela = tibble::tibble(x=c(4,0,6), y=c(0,1,4)) #cria a tabela de pontos
2    ggplot2::ggplot()+ #cria uma janela gráfica
3    geom_point(data=tabela, aes(x=x, y=y), color='black') #cria os pontos na janela gráfica
```

O detalhamento do código é feito a seguir:

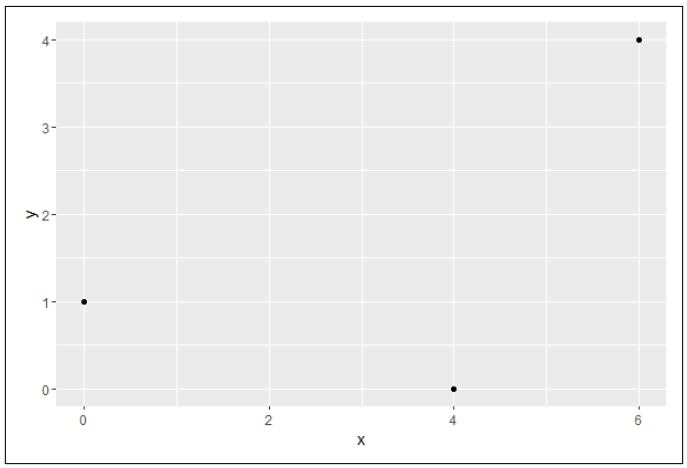
1. Cria com o pacote *tibble* uma tabela com duas colunas. Após # comentário;

2. Cria uma janela gráfica com o pacote *ggplot2*, o sinal de + indica inclusão de nova camada nesta janela, informada na próxima linha;

3. Acrescenta uma camada de pontos cujos dados estão na tabela; define que a cor dos pontos será *black*.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 2.1: Exemplo de gráfico de pontos com o pacote *ggplot2*.



Fonte: Os autores.

Conforme o leitor poderá constatar, os código de visualização das curvas e mandalas que serão abordados aqui são limitados ao formato de pontos, no entanto deve ser destacado que o pacote *ggplot2* possui inúmeras opções, cujo detalhamento fogem ao escopo deste e-book.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Capítulo 3

Conceitos de Matemática e Programação

Neste capítulo são apresentadas algumas noções gerais sobre curvas planas, diferentes formas de caracterizá-las, além de uma apresentação mais detalhada das curvas que serão utilizadas para a geração das figuras do capítulo 6. Este capítulo será iniciado com uma pequena discussão sobre curvas planas e, em seguida, um conjunto específico de curvas é apresentado em forma paramétrica. Para o leitor interessado em maiores detalhes sobre cada uma das curvas, as referências podem ser um ponto inicial interessante.

O conjunto de curvas planas utilizado é composto pelo circunferência, a elipse, o cardioide, deltoide, o astroide, lemniscata de Bernoulli, lemniscata de Gerono e o limaçon de Pascal.

3.1 Curvas Planas

Uma curva, intuitivamente, pode ser considerada como uma trajetória descrita pelo movimento de uma partícula que deixa um rastro à medida que se move. Fazendo traços com um lápis em um papel, desenha-se curvas que podem ser vistas como o rastro das trajetórias do ponto de contato do lápis com o papel. Essa é apenas uma ideia intuitiva do que é uma curva, não uma definição. Do ponto de vista da matemática, uma curva pode ser caracterizada como um lugar geométrico no plano. Um exemplo simples é a circunferência: lugar geométrico dos pontos do plano que são equidistantes a um ponto fixado, o centro da circunferência. No entanto, para implementações computacionais, é necessário utilizar a geometria analítica que, por meio da adoção de um sistemas de coordenadas, permite expressar objetos geométricos tais como pontos, vetores e curvas por meio de coordenadas e equações, permitindo assim sua implementação computacional. Como ilustração, uma circunferência de centro $(0, 0)$ e raio r pode ser representada:

i.) implicitamente, em coordenadas cartesianas, pela equação $x^2 + y^2 = r^2$;

ii.) explicitamente, em oordenadas cartesianas, como reunião dos gráficos de $y = \sqrt{r^2 - x^2}$,

para $|x| \leq r$ na parte superior e $y = -\sqrt{r^2 - x^2}$, para $|x| \leq r$ na parte inferior;

iii.) explicitamente, em coordenadas polares (ρ, θ), pela equação $\rho = r, 0 \leq \theta \leq 2\pi$;

iv.) parametricamente, por $\begin{cases} x = r \cdot \cos t \\ y = rr \cdot \operatorname{sen} t \end{cases}$, $t \in [0, 2\pi]$.

Como no caso da circunferência, curvas planas podem ser caracterizadas das diversas formas descritas anteriormente: como um lugar geométrico, por uma equação, como o gráfico de uma função de uma variável, por uma parametrização.

Desenhar curvas no computador requer o uso de um *software* apropriado. Neste texto, utilizando a linguagem R e o pacote *ggplot2*, as curvas são geradas à partir de uma tabela de valores, tratada como um conjunto de dados. Por essa razão, dentre as diversas formas de representação de uma curva, computacionalmente, a melhor forma é a paramétrica, em que as coordenadas de cada ponto da curva são escritas na forma $(x(t), y(t))$ e o parâmetro t varia em um conjunto pré definido de valores. Isso deve-se à facilidade para gerar a tabela de valores que será usada como conjunto de dados. Curvas dadas como gráficos de funções são curvas parametrizadas em que os pontos são da forma $(x, f(x))$ em que x, a variável independente, é o parâmetro. Por essa razão, todos os códigos em R apresentados neste texto usam curvas parametrizadas. Eventualmente, as expressões em coordenadas polares são fornecidas.

3.2 Circunferência

Uma das curvas mais conhecidas, antigas e fundamentais da história da humanidade. Segundo [16], as propriedades do circunferência são componentes fundamentais para a invenção da roda. A história registrada é anterior ao escriba de *Ahmes*, o qual fornece para o cálculo da área do círculo um valor de $\pi \approx 3.16$.

O círculo desempenhou um papel fundamental no estudo das curvas clássicas, pois várias destas estão relacionadas ao problema de quadratura do círculo [16]. Uma leitura interessante sobre a história do π é a referência [17]. A simplicidade das equações da circunferência e respectivo traçado, podem obscurecer as possibilidades de utilização para a produção de figuras geométricas e simétricas. Em coordenadas paramétricas:

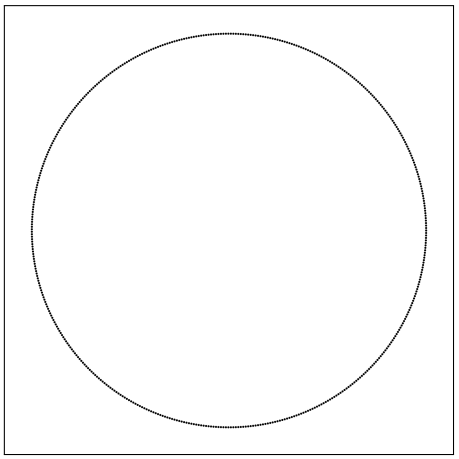
$$x = r \cdot \operatorname{sen}(t), \qquad y = r \cdot \cos(t), \qquad t \in [0, 2\pi].$$

A Figura 3.1 e respectivo código em R são mostrados a seguir:

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 3.1: Circunferência.

```
require(ggplot2)
n=500
t=seq(0,2*pi, length.out = n)
raio=1
x=raio*cos(t); y=raio*sin(t)
dt=tibble::tibble(x,y)
p= ggplot()+  coord_fixed()+theme_void()
p=p+ geom_point(data=dt, aes(x=x, y=y),
color='black')
p
```



Fonte: Os autores.

O detalhamento do código é apresentado a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos.

3. Calcula a sequência de pontos t no intervalo $[0, 2\pi]$.

4. Define o raio do circunferência adotado.

5. Cálculo dos valores x, y das coordenadas paramétricas do circunferência.

6. Constrói o banco de dados de pontos (x, y).

7.-9. Elementos gráficos.

3.3 Elipse

A elipse é definida como o lugar geométrico dos pontos $P(x, y)$ tais que a soma das distâncias a dois pontos F_1 e F_2, denominados focos, é uma constante. Segundo [18], foi estudada primeiramente por Menaechmus e suas aplicações não se restringem ao estudo das cônicas, mas ramificam-se pelas aplicações em astronomia. Informações adicionais e algumas referências podem ser encontradas em [18] e [28].

A Equação reduzida em coordenadas cartesianas (considerando centro $(0, 0)$ e eixo focal coincidente com um dos eixos coordenados) é dada por:

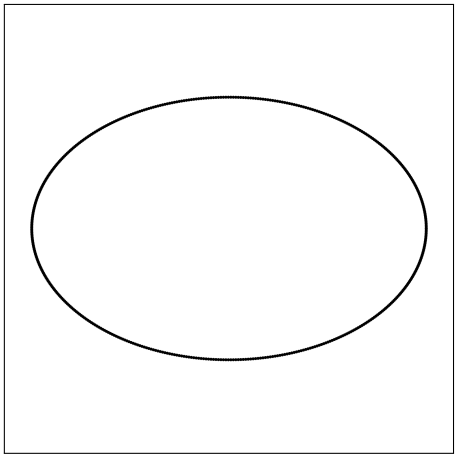
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

As equações paramétricas são dadas por:

$$x = a \cdot \cos(t), \qquad y = b \cdot \text{sen}(t), \qquad t \in [0, 2\pi], \quad a, b : constantes.$$

Figura 3.2: Elipse.

```
require(ggplot2)
n=500
theta=seq(0,2*pi,pi/314)
a = 2; b=1
x = a*cos(theta);  y = b*sin(theta)
dt=tibble::tibble(x,y)
p= ggplot()+  coord_fixed()+theme_void()
p=p+ geom_point(data=dt, aes(x=x, y=y),
color='black')
p
```



Fonte: Os autores.

O detalhamento do código é apresentado a seguir:

1. Carrega o pacote *ggplot2*;

2. Define a quantidade de pontos n a ser utilizada para os cálculos;

3. Calcula a sequência de pontos t no intervalo $[0, 2\pi]$;

4. Define os eixos maior **a** e menor **b**;

5. Cálculo dos valores x, y das coordenadas paramétricas da elipse;

6. Construindo o banco de dados de pontos (x, y);

7.-9. Elementos gráficos.

3.4 Cardioide

Um cardioide é uma curva cuja forma se assemelha à representação de um coração. Por este motivo, recebe o nome derivado do grego kardioeides = kardia: coração + eidos: forma. O nome cardioide é creditado a Castillon em *Philosophical Transactions of the Royal Society* em 1741 [29]. Considerando duas circunferências de mesmo raio, tangentes em um ponto P, é a curva descrita por P quando uma circunferência gira (sem deslizar) tangencialmente sobre a outra circunferência (mantida fixa) [19]. Raios distintos para as circunferências produzem uma curva mais geral denominada epicicloide [27].

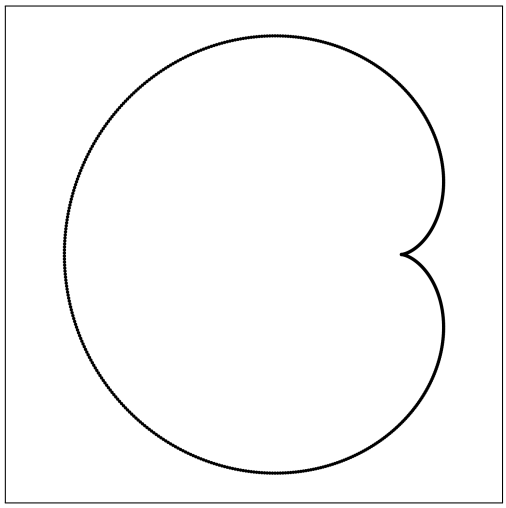
Uma parametrização do cardioide é dada por [28].

$$x = r \cdot (1 - \cos(\theta)) \cdot \cos(\theta), \quad y = r \cdot (1 - \cos(\theta)) \cdot \operatorname{sen}(\theta), \quad \theta \in [0, 2\pi] \tag{3.4.1}$$

Por questão de conveniência, a parametrização (3.4.1) será utilizada. A Figura 3.3 mostra o cardioide e respectivo código.

Figura 3.3: Cardioide.

```
require(ggplot2)
n=700
theta=seq(0,2*pi, length.out = n)
raio=1
x=(1-cos(theta))*cos(theta)*raio;
y=(1-cos(theta))*sin(theta)*raio
dt=tibble::tibble(x,y)
size=0.5
p=ggplot()+coord_fixed()+theme_void()
p= p+geom_point(data=dt, aes(x=x, y=y),
color='black',size=size)
p
```



Fonte: Os autores.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos.

3. Calcula a sequência de pontos θ no intervalo $[0, 2\pi]$.

4. Define o raio.

5.-6. Calcula aos valores x e y.

7. Constrói banco de dados de pontos (x, y).

8.-12. Parâmetros gráficos.

3.5 Deltoide

O deltoide é uma curva tricúspide e foi estudada por Euler em 1745 e por Steiner em 1856 [30]. É um caso particular de uma hipocicloide que é a curva descrita pelo ponto de

tangência P entre duas circunferências, uma interna à outra, quando a circunferência interna gira (sem deslizar) sobre a circunferência externa (mantida fixa). Quando a razão entre os raios da circunferência externa e da circunferência interna é 3, a curva descrita pelo ponto P é a deltoide.

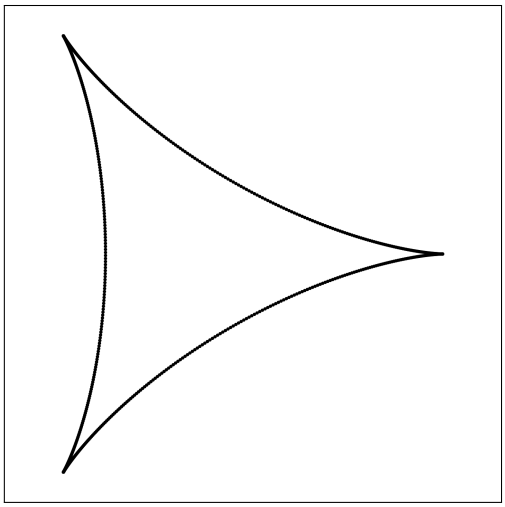
As equações paramétricas são dadas por:

$$x = a \cdot (2\cos(t) + \cos(2t)) \qquad y = a \cdot (2\,\mathrm{sen}(t) - \mathrm{sen}(2t)), \qquad t \in [0, 2\pi].$$

A seguir estão o código e a respectiva Figura 3.4.

Figura 3.4: Deltoide.

```
require(ggplot2)
n=700
theta=seq(0,2*pi, length.out = n)
a=1
x=a*(2*cos(theta)+cos(2*theta))
y=a*(2*sin(theta)-sin(2*theta))
dt=tibble::tibble(x,y)
size=0.5
p = ggplot()+coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=x, y=y),
color='black',size=size)
p
```



Fonte: Os autores.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Define o valor do parâmetro a.

5-6. Calcula aos valores x e y.

7. Constrói banco de dados de pontos (x, y).

8.-12. Parâmetros gráficos.

3.6 Astroide

O astroide é uma curva tetracúspide e foi estudada , primeiramente, por Johann Bernoulli in 1691-92 e Leibniz em 1715 [21]. Também é um caso particular da hipocicloide. Neste caso,

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

a razão entre os raios da circunferência externa e da circunferência interna é igual a 4.

Uma descrição mais detalhada e referências podem ser encontradas em [31].

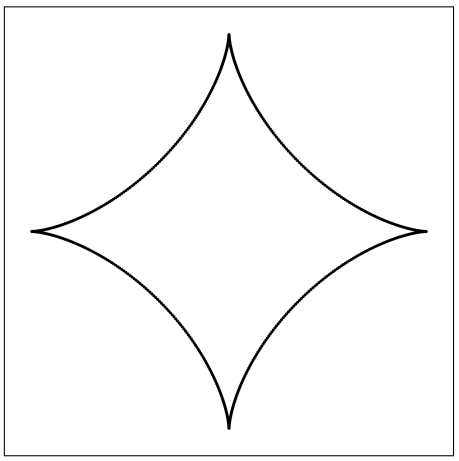
As equações paramétricas[21] são dadas por:

$$x = a \cdot \cos^3(t) \qquad y = a \cdot \operatorname{sen}^3(t), \qquad t \in [0, 2\pi],$$

A seguir estão o código e a respectiva figura 3.5 do astroide.

Figura 3.5: Astroide.

```
require(ggplot2)
n=500
thetaa=seq(0,2*pi, length.out = n)
a=1
x=a*(cos(theta))^3; y=a*(sin(theta))^3
dt=tibble::tibble(x,y)
size=0.5
p = ggplot()+coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=x, y=y),
color='black',size=size)
p
```



Fonte: Os autores.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Define o valor do parâmetro a.

5. Calcula aos valores x e y.

6. Constrói banco de dados de pontos (x, y).

7.-11. Parâmetros gráficos.

3.7 Lemniscata de Bernoulli

A Lemniscata é uma figura geométrica em forma de hélice que surgiu em 1694 com Jacob Bernoulli e é um caso especial de um conjunto de curvas mais gerais. Ver, por exemplo, [5].

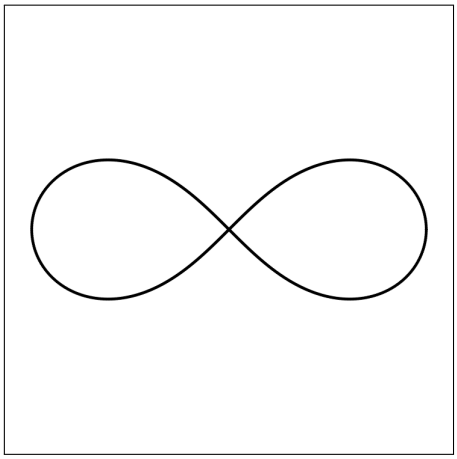
As equações paramétricas são dadas por [32]:

$$x = \frac{a\cos(t)}{1+\text{sen}^2(t)}, \quad y = \frac{a\cos(t)\,\text{sen}(t)}{1+\text{sen}^2(t)}, t \in [0, 2\pi].$$

A seguir estão apresentados o código e respectiva Figura 3.6 da Lemniscata de Bernoulli.

Figura 3.6: Lemniscata.

```
1   require(ggplot2)
2   n=700
3   theta=seq(0,2*pi, length.out = n)
4   x=cos(theta)/(1+sin(theta)^2);
5   y=sin(theta)*cos(theta)/(1+sin(theta)^2)
6   dt=tibble::tibble(x,y)
7   size=0.5
8   p = ggplot()+coord_fixed()+theme_void()
9   p = p+geom_point(data=dt, aes(x=x, y=y),
10  color='black',size=size)
11  p
```



Fonte: Os autores.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2.-3. Define a quantidade de pontos n a ser utilizada para os cálculos e a sequência de pontos do parâmetro `theta` $\in [0, 2\pi]$.

4.-5. Cálculo dos valores x e y das coordenadas.

6. Constrói o banco de dados de pontos (x, y).

7.-11. Elementos gráficos.

3.8 Leminiscata de Gerono

A Lemniscata de Gerono é uma figura similar à Lemniscata de Bernoulli, no entanto, por ser mais achatada a sua equação é consideravelmente diferente. A equação é dada por [33]:

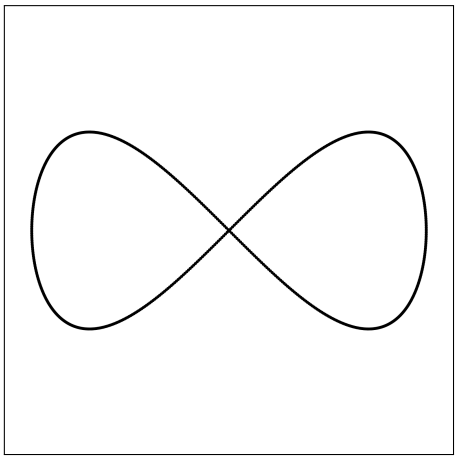
$$x = \text{sen}(\theta), \qquad y = \cos(\theta)\,\text{sen}(\theta), \quad \theta \in [0, 2\pi].$$

A seguir estão apresentados o código e respectiva Figura 3.7 da Lemniscata de Gerono.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 3.7: Lemniscata.

```
require(ggplot2)
n=700;
theta=seq(0,2*pi, length.out = n)
x=sin(theta);
y=sin(theta)*cos(theta)
dt=tibble::tibble(x,y)
size=0.5
p = ggplot()+coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=x, y=y),
color='black',size=size)
p
```



Fonte: Os autores.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2.-3. Define a quantidade de pontos n a ser utilizada para os cálculos e a sequência de pontos do parâmetro `theta` $\in [0, 2\pi]$.

4.-5. Cálculo dos valores x e y das coordenadas.

6. Constrói o banco de dados de pontos (x, y).

7.-11. Elementos gráficos.

3.9 Limaçon de Pascal

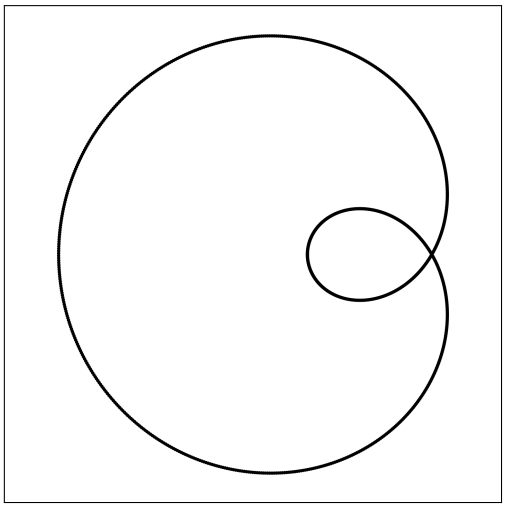
A curva Limaçon de Pascal foi descoberta pelo pai de Blaise Pascal, Étienne Pascal. Seu nome é creditado ao francês Gilles-Personne Roberval in 1650 [20]. Possui o cardioide como um caso especial e equações paramétricas dadas por [34]:

$$\begin{aligned} x &= (b + a \cdot \cos(t)) \cdot \cos(t) \\ y &= (b + a \cdot \cos(t)) \cdot \operatorname{sen}(t) \end{aligned}, \qquad t \in [0, 2\pi]$$

A seguir estão o código e a Figura 3.8 do Limaçon de Pascal. O sinal positivo foi trocado pelo sinal negativo apenas para obter a cúspide a direita da figura.

Figura 3.8: Limaçon de Pascal.

```
require(ggplot2)
n=1000
t=seq(0,2*pi, length.out = n)
b=1;a=2
x=(b-a*cos(t))*cos(t)
y=(b-a*cos(t))*sin(t)
dt=tibble::tibble(x,y)
size=0.5
p = ggplot()+coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=x, y=y),
color='black',size=size)
p
```



Fonte: Os autores.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos.

3. Calcula a sequência de pontos t no intervalo $[0, 2\pi]$.

4. Define os valores dos parâmetros a b.

5.-6. Calcula aos valores x e y.

7. Constrói banco de dados de pontos (x, y).

8.-12. Parâmetros gráficos.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Capítulo 4

Transformações Geométricas com R

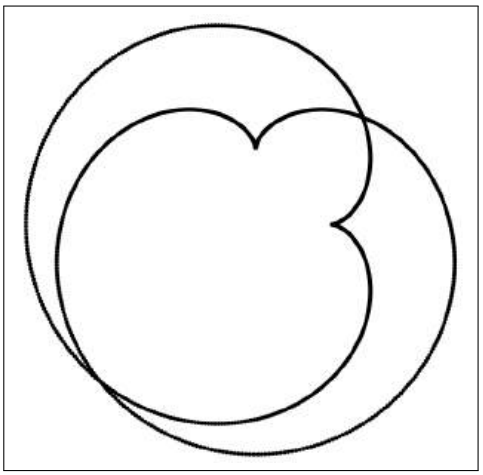
Neste capítulo, as transformações geométricas de rotações e homotetias são aplicadas aos gráficos das curvas mostradas no capítulo 3. A aplicação adequada desses movimentos rígidos de rotação e homotetia tornam possível combinar figuras de modo que o resultado possua aspectos simétricos e, eventualmente, um entrelaçamento de linhas que proporciona um visual interessante. Por razão de comodidade e simplificação, todas as transformações rígidas são realizadas em relação a origem do sistema de coordenadas cartesianas.

4.1 Rotação do Cardioide

A Figura 4.1 mostra o cardioide original e o cardioide rotacionado por um ângulo $\theta = \pi/2$.

```
require(ggplot2)
n=700
theta=seq(0,2*pi, length.out = n)
r=1
x=c(2*r*cos(theta)-r*cos(2*theta))
y=c(2*r*sin(theta)-r*sin(2*theta))
rotacao=pi/2
xr=x;   yr=y
xr=c(xr,x*cos(rotacao)-y*sin(rotacao))
yr=c(yr,x*sin(rotacao)+y*cos(rotacao))
dt=tibble::tibble(xr,yr)
size=0.5
p = ggplot()+coord_fixed()+theme_void()
p = p+geom_point(data=dt,aes(x=xr, y=yr),
color='black',size=size)
p
```

Figura 4.1: Rotação do Cardioide.



Fonte: Os autores.

Note que a base de dados *dt* contém todos os pontos, ou seja, todos os pontos (x, y) da figura original e todos os pontos (x, y) obtidos por meio da rotação. O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e o raio.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Define o raio.

5. Calcula aos valores x.

6. Calcula aos valores y.

7. Define o ângulo de rotação.

8. Faz cópia dos valores de x e y.

10. Calcula aos valores xr utilizando a fórmula de rotação.

11. Calcula aos valores yr utilizando a fórmula de rotação.

12.-16. Parâmetros gráficos.

Note que apenas o código para a rotação foi inserido e que não há mais modificações. Qualquer outra curva pode ser utilizada no lugar do cardioide sem maiores modificações do código.

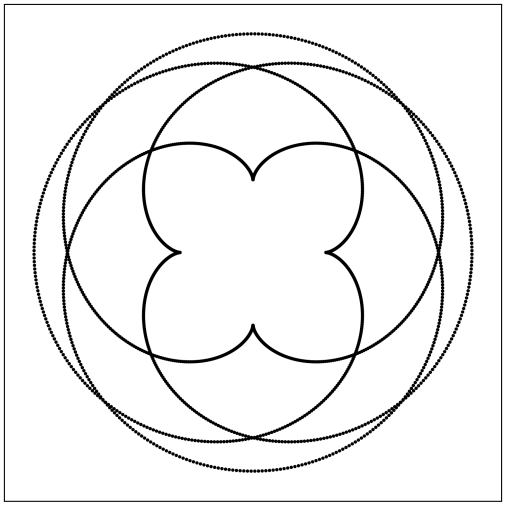
4.2 Rotações Sucessivas

Vamos realizar rotações sucessivas do cardioide por um ângulo θ, escolhido arbitrariamente, e fazer a composição para obter a Figura 4.2. Nesse caso, a abordagem anterior pode ser executada, ou seja, a definição manual de cada uma das rotações, porém, o processo é ineficiente do ponto de vista de repetição do código e pode ser melhor abordado por meio de uma estrutura de fluxo do tipo *loop*.

A Figura 4.2 e respectivo código são mostrados a seguir. Note que: i.) $rotacao = c(pi/2, pi, 3 * pi/2)$ agora é um vetor que armazena os ângulos a serem utilizados no processo; ii.) o *loop for* executa iteração sobre cada um dos elementos de *rotacao* calculando as rotação do ângulo na respectiva posição i; iii.) o resultado do item ii.) é concatenado aos resultados previamente calculados até o fim do processo.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 4.2: Composição de rotações do cardioide.



Fonte: Os autores.

```
require(ggplot2)
n=500; raio=1
theta=seq(0,2*pi, length.out = n)
x=c(2*raio*cos(theta)-raio*cos(2*theta))
y=c(2*raio*sin(theta)-raio*sin(2*theta))
rotacao=c(pi/2,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]))
}
dt=tibble::tibble(xt,yt)
size=0.5
p = ggplot()+   coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=xt, y=yt), color='black',size=size)
p
```

A descrição do código é idêntica ao caso anterior, exceto pela inserção do *loop* para as rotações sucessivas. Os detalhes são apresentados a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e o raio.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Calcula aos valores x.

5. Calcula aos valores y.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

6. Define os ângulos de rotação.

7. Faz cópia dos valores de x e y.

8.-11. Executa cálculos sucessivos dos valores xt e yt utilizando a fórmula de rotação de ponto.

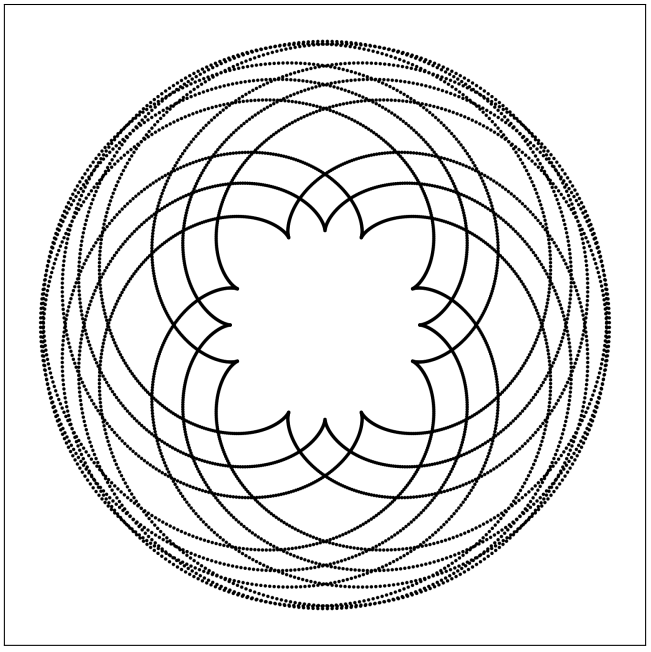
12. Constrói o banco de dados.

13.-16. Parâmetros gráficos.

4.3 Rotações Sucessivas Anti-horárias e Horárias

Nesta seção, o cardioide é rotacionado por ângulos $rotacao = c(pi/2, pi, 3 * pi/2)$. Em seguida, a composição é rotacionada de ângulo de $\pi/8$ radianos em ambos os sentidos anti-horário e horário. O código e a Figura 4.3, resultantes do processo de composição, são mostrados a seguir:

Figura 4.3: Rotações anti-horárias e horárias.



Fonte: Os autores.

Note que a composição o cardioide com cúspide em ângulo zero é rotacionada em $\pi/2, \pi, 3\pi/2$ para obter os cardioides com cúspides em $\pi/2, \pi, 3\pi/2$, respectivamente. Todas as rotações fornecem uma figura composta $F0$. A rotação em $\pi/8$ de $F0$ fornece as cúspides em $\pi/8, \pi/2 + \pi/8, \pi + \pi/8, 3\pi/2 + \pi/8$, enquanto que a rotação em $-\pi/8$ fornece as cúspides em $-\pi/8, \pi/2 - \pi/8, \pi - \pi/8, 3\pi/2 - \pi/8$. Esse processo de rotações horárias e anti-horárias pode ser repetido

para um vetor qualquer, como por exemplo, $rot = c(seq(0, 2 * pi, pi/8), rotacao = c(rot, -rot)$ produzirá uma rotação de ângulos múltiplos horários e anti-horários de $pi/8$.

```
require(ggplot2)
n=1500;  raio=1
theta=seq(0,2*pi, length.out = n)
x=c(2*raio*cos(theta)-raio*cos(2*theta))
y=c(2*raio*sin(theta)-raio*sin(2*theta))
rotacao=c(pi/2,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]))
}
rotacao=c(pi/8,pi,-pi/8)
xt1=c(xt*cos(rotacao)-yt*sin(rotacao))
yt1=c(xt*sin(rotacao)+yt*cos(rotacao))
dt=tibble::tibble(xt1,yt1)
size=0.25
p = ggplot()+   coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=xt1, y=yt1), color='black',size=size)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e o raio.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Calcula aos valores x.

5. Calcula aos valores y.

6. Define os ângulos de rotações.

7. Faz cópia dos valores de x e y.

8.-11. Executa cálculos sucessivos dos valores xt e yt utilizando a fórmula de rotação.

12.-15. Calcula rotação da figura gerada anteriormente pelos ângulos dados. Constrói o banco de dados.

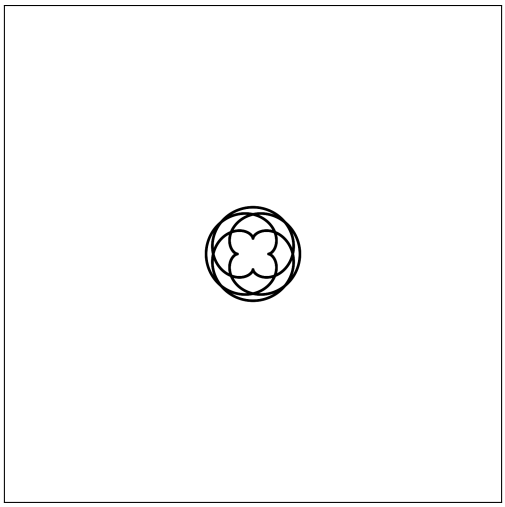
15.-19. Parâmetros gráficos.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

4.4 Homotetia

Nesta parte, uma homotetia de razão $c = 0.25$ é aplicada à composição das rotações mostrada na Figura 4.2. O resultado é mostrado na Figura 4.4.

Figura 4.4: Homotetia de razão $c = 0.25$ da composição de rotações da Figura 4.2.



Fonte: Os autores.

Os códigos são apresentados a seguir:

```
require(ggplot2)
n=1000; raio=1
theta=seq(0,2*pi, length.out = n)
x=c(2*raio*cos(theta)-raio*cos(2*theta))
y=c(2*raio*sin(theta)-raio*sin(2*theta))
rotacao=c(pi/2,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]))
}
contracao = 0.25
xt2=xt*contracao;  yt2=yt*contracao
dt2=tibble::tibble(xt2,yt2)
size=0.25
p = ggplot()+   coord_fixed()+theme_void()
p = p+geom_point(data=dt2, aes(x=xt2, y=yt2), color='black',size=size)+
scale_y_continuous(limits = c(-3.5,3.5))+ scale_x_continuous(limits = c(-3.5,3.5))
p
```

O detalhamento é idêntico ao anterior, exceto pela definição dos limites dos eixos para evitar o escalonamento automático.

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e o raio.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Calcula aos valores x.

5. Calcula aos valores y.

6. Define os ângulos de rotações.

7. Faz cópia dos valores de x e y.

8.-11. Executa cálculos sucessivos dos valores xt e yt utilizando a fórmula de rotação.

12. Define a razão de homotetia.

13. Calculo dos pontos $xt2$ e $yt2$, os pontos com homotetia.

14. Define o banco de dados.

15.-19. Parâmetros gráficos.

4.5 Composição de rotações e homotetias

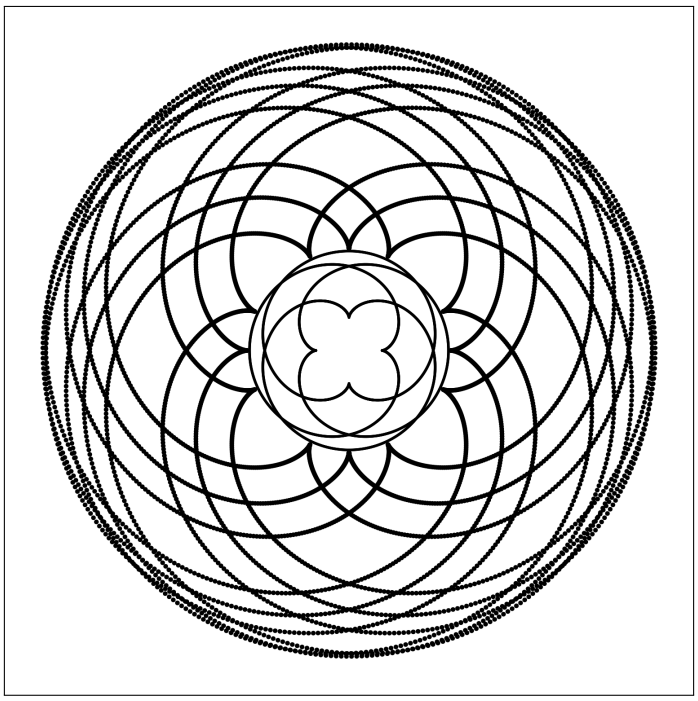
Nesta seção, os elementos discutidos nas seções 4.2, 4.3 e 4.4 são concatenados em um única figura, ou seja, a figura é gerada pela composição de rotações horária, anti-horária e homotetia. O procedimento consiste em:

- Construir o cardioide.
- Rotacionar o cardioide no sentido horário para obter a composição de rotações.
- Rotacionar a composição anterior nos sentidos horário e anti-horário e construir a composição.
- Realizar a homotetia da composição anterior.
- Compor a figura final.

A Figura 4.5 mostra a composição entre as rotações horária e anti-horária do cardioide e a homotetia. A razão de homotetia para fornecer o resultado apresentado foi determinada manualmente por inspeção.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 4.5: Composição de rotações, anti-horária e horária, e homotetia de razão $c = 0.325$.



Fonte: Os autores.

```
require(ggplot2)
n=1000;        raio=1
theta=seq(0,2*pi, length.out = n)
x=c(2*raio*cos(theta)-raio*cos(2*theta))
y=c(2*raio*sin(theta)-raio*sin(2*theta))
rotacao=c(pi/2,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]))
}
rotacao=c(pi/8,pi,-pi/8)
xt1=c(xt*cos(rotacao)-yt*sin(rotacao))
yt1=c(xt*sin(rotacao)+yt*cos(rotacao))
dt1=tibble::tibble(xt1,yt1)
contracao = 0.325
xt2=xt*contracao; yt2=yt*contracao
dt2=tibble::tibble(xt2,yt2)
size=0.5
p= ggplot()+ coord_fixed()+ theme_void()
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color='black',size=size)+
geom_point(data=dt2, aes(x=xt2, y=yt2), color='black',size=0.015)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e o raio.

3. Calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

4. Calcula aos valores x.

5. Calcula aos valores y.

6. Define os ângulos de rotações.

7. Faz cópia dos valores de x e y.

8.-11. Executa cálculos sucessivos dos valores xt e yt utilizando a fórmula de rotação.

12. Define aos ângulos de rotação nos sentido horário e anti-horário.

13.-14. Calculo dos pontos $xt1$ e $yt1$: os pontos rotacionados.

15. Define o banco de dados de $xt1$ e $yt1$.

16. Define a razão de homotetia.

17.-18. Calculo dos pontos $xt2$ e $yt2$ com a razão de homotetia e construção do banco de dados.

19.-23. Parâmetros gráficos.

As ideias expressas para a composição da figura anterior, são desenvolvidas para as próximas construções. De forma geral, o mesmo procedimento é aplicado. No entanto, é necessário desenvolver, modificar ou alterar partes do código para que a construção possa ser feita, pois há detalhes inerentes às figuras base adotadas para a construção.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Capítulo 5

Mandalas

Neste capítulo são apresentados algumas figuras interessantes provenientes da utilização de algumas curvas planas. As curvas selecionadas refletem um aspecto pessoal, mas não há restrições quanto a utilização de outras curvas. Aqui, todas as figuras são deixadas em preto, pois a ênfase está no processo de construção utilizando o R. Uma discussão sobre coloração é apresentada no capítulo 6.

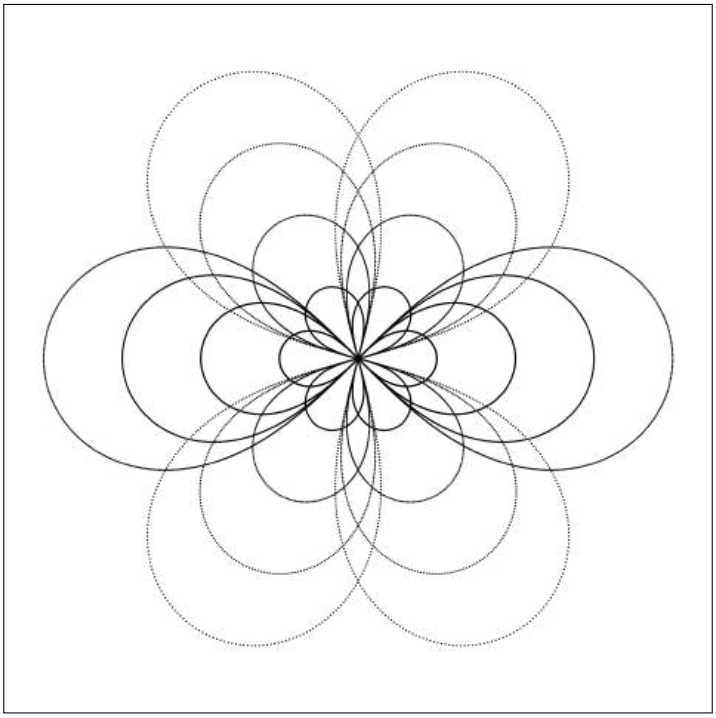
5.1 Lemniscata de Bernoulli

Nesta seção, um conjunto de rotações sucessivas da Lemniscata de Bernoulli perfazem a primeira parte do processo de composição. Em seguida são realizadas homotetias e respectivas composições com os resultados iniciais. A Figura 5.1 ilustra o resultado do procedimento construtivo descrito a seguir:

- Construção da Lemniscata de Bernoulli.
- Rotações sucessivamente por ângulos $\theta = 0, \frac{\pi}{3}, \frac{2\pi}{3}$.
- Obtenção dos vetor x_t, y_t contendo todos os valores em x e y rotacionados.
- Homotetia de razão $\frac{1}{4}$, $\frac{1}{2}$ e $\frac{3}{4}$ para x_t, y_t.
- Composição da figura com os resultados gerados nos itens anteriores.

A Figura 5.1 deixa em destaque a lemniscata de Bernoulli que pode ser vista como uma rotação de ângulo zero ou ângulo π. As demais rotações de ângulos $\pi/3$ e $2\pi/3$ produzem as outras duas lemniscatas. Quaisquer outros ângulos podem ser escolhidos, bastando redefinir o vetor *rotacao*. Um ângulo de $\pi/2$ produz uma figura parece com uma figura de pétala, enquanto que muitas rotações ocasionam sobreposições. A Figura 5.2 ilustra um caso de sobreposições.

Figura 5.1: Composição de lemniscatas rotacionadas e homotetias.



Fonte: Os autores.

O código completo é apresentado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n)
x=cos(theta)/(1+(sin(theta))^2); y=sin(theta)*cos(theta)/(1+(sin(theta))^2)
rotacao=c(seq(0,pi,pi/3))
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 = 0.75; xt4=xt*contracao; yt4=yt*contracao
dt4=tibble::tibble(xt4,yt4); size=0.025
p= ggplot()+ coord_fixed()+ theme_void()
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color='black',size=size)
p=p+geom_point(data=dt2, aes(x=xt2, y=yt2), color='black',size=size)
p = p+geom_point(data=dt3, aes(x=xt3, y=yt3), color='black',size=size)
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color='black',size=size)
```

```
22          p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3. Calcula aos valores x e y.

4. Define os ângulos de rotações.

5. Faz cópia dos valores de x e y.

6.-10. Calcula xt e yt utilizando a fórmula de rotação. Constrói o banco de dados.

11.-16. Define a razão da homotetia e calcula os respectivos $xt2$ e $yt2$. Constrói banco de dados.

17.-22. Parâmetros gráficos.

5.2 Lemniscata de Gerono

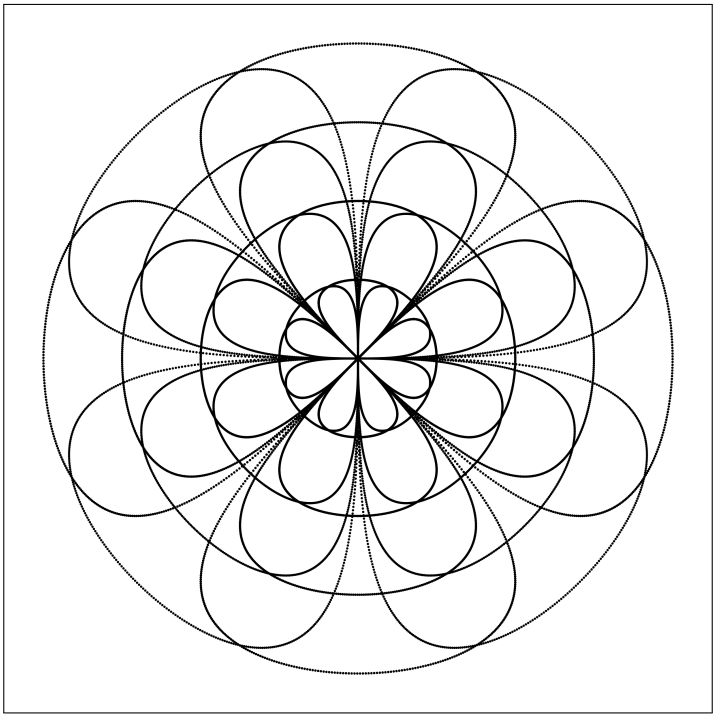
Nesta seção, um conjunto de rotações sucessivas da Lemniscata de Gerono perfazem a primeira parte do processo de composição. Em seguida são realizadas homotetias e respectivas composições com os resultados inciais. A Figura 5.2 ilustra o resultado do procedimento construtivo descrito a seguir:

- Construção da lemniscata de Gerono.
- Rotações sucessivamente por ângulos $\theta = \frac{\pi}{4}, \frac{\pi}{2}, 3 \cdot \frac{\pi}{4}, \pi, 3 \cdot \frac{\pi}{2}$.
- Obtenção dos vetor x_t, y_t contendo todos os valores em x e y rotacionados.
- Homotetia de razão $\frac{1}{4}$, $\frac{1}{2}$ e $\frac{3}{4}$ para x_t, y_t.
- Composição da figura com os resultados gerados nos itens anteriores.

A Figura 5.2 ilustra o caso em que as rotações de ângulo zero e *pi* foram descartadas. Deve ser notado que as pétalas aparentes não são as lemniscatas, mas resultados das interseções.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 5.2: Composição de lemniscatas rotacionadas e homotetias.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta)
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]))}
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 = 0.75; 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='black',size=size)
p=p+geom_point(data=dt2, aes(x=xt2, y=yt2), color='black',size=size)
p = p+geom_point(data=dt3, aes(x=xt3, y=yt3), color='black',size=size)
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color='black',size=size)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3. Calcula aos valores x e y.

4. Define os ângulos de rotações.

5. Faz cópia dos valores de x e y.

6.-10. Calcula xt e yt utilizando a fórmula de rotação. Constrói o banco de dados.

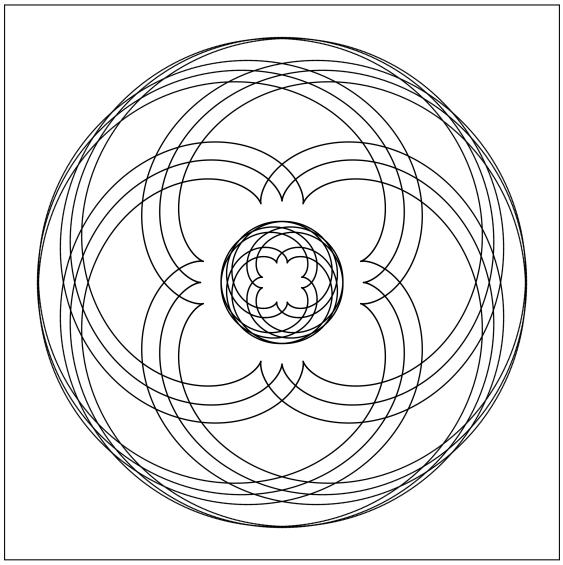
11.-16. Define a razão da homotetia e calcula os respectivos $xt2$ e $yt2$. Constrói banco de dados.

17.-23. Parâmetros gráficos.

5.3 Cardioide

Um cardioide é utilizado como base principal e, em seguida, três rotações sucessivas perfazem a primeira parte da composição. Rotações horárias e anti-horárias finalizam os cálculos. A Figura 5.3 ilustra um resultado similar ao mostrado na seção 4.5, com diferenciação no ângulo das rotações e na razão da homotetia.

Figura 5.3: Composição de cardioides com rotações e homotetias.



Fonte: Os autores.

Note que a Figura 5.3 é uma pequena modificação da Figura 4.5. Essa diferença pode ser observada no código completo fornecido a seguir:

```
require(ggplot2)
n=1500; theta=seq(0,2*pi, length.out = n);  raio=1;
x=c(2*raio*cos(tetha)-raio*cos(2*tetha))
y=c(2*raio*sin(tetha)-raio*sin(2*tetha))
dt=tibble::tibble(x,y)
rotacao=c(pi/2,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]))
}
dt=tibble::tibble(xt,yt)
rotacao = c(pi/12,-pi/12);  xt1=NULL; yt1=NULL
n=length(xt)
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]))
}
dt1=tibble::tibble(xt1,yt1)
contracao = 0.25; xt2=xt1*contracao; yt2=yt1*contracao
dt2=tibble::tibble(xt2,yt2)
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=xt, y=yt), color='black',size=.05)
p=p+geom_point(data=dt1, aes(x=xt1, y=yt1), color='black',size=.05)+
geom_point(data=dt2, aes(x=xt2, y=yt2), color='black',size=.025)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$ e o raio.

3.-5. Calcula aos valores x e y. Constrói banco de dados.

6. Define os ângulos de rotações e faz cópia dos valores de x e y.

7.-11. Calcula xt e yt utilizando a fórmula de rotação. Constrói o banco de dados.

12.-18. Define ângulos de rotação, comprimento do vetor xt e rotaciona xt por ângulos no vetor de rotação. Constrói banco de dados $dt1$.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

19.-20. Define a razão da homotetia e calcula os respectivos $xt2$ e $yt2$. Constrói banco de dados $dt2$.

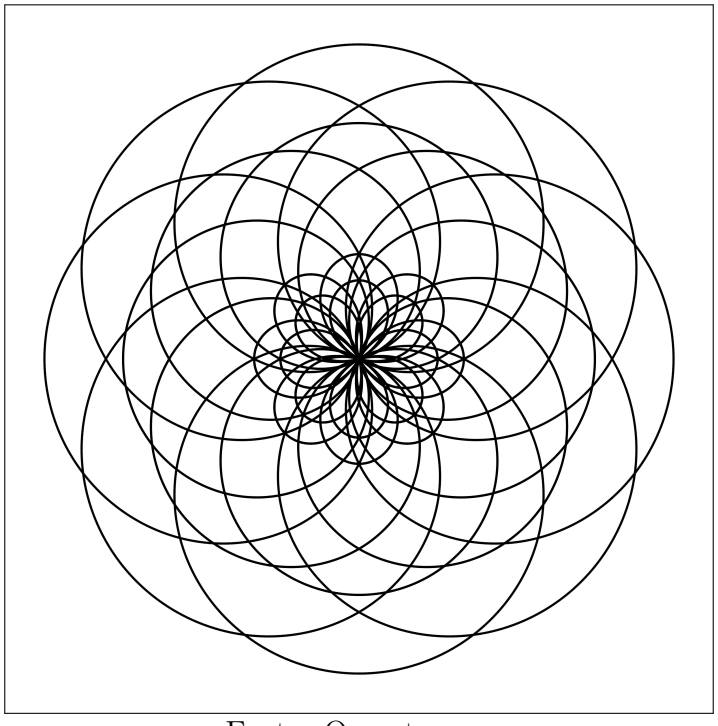
21.-25. Parâmetros gráficos.

5.4 Limaçon de Pascal

Neste caso, um Limaçon é utilizado como base principal e, em seguida, rotações sucessivas perfazem a primeira parte da composição. Homotetias finalizam os cálculos. A Figura 5.4 ilustra o resultado do procedimento construtivo a seguir:

- Construção do Limaçon de Pascal.
- Rotações sucessivamente por ângulos $\theta = \frac{\pi}{4}, \frac{\pi}{2}, 3 \cdot \frac{\pi}{4}, \pi, 5 \cdot \frac{\pi}{4}, 6 \cdot \frac{\pi}{4}, 7 \cdot \frac{\pi}{4}$.
- Cálculo de x_t e y_t com todos os valores em x e y rotacionados.
- Homotetias de razão $0,25$, $0,5$ e $0,75$ para x_t, y_t.
- Composição da figura com os resultados gerados.

Figura 5.4: Composição de rotações e homotetias do Limaçon de Pascal.



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=1500; theta=seq(0,2*pi, length.out = n)
x=cos(theta)*(2*cos(theta)+1)
y=sin(theta)*(2*cos(theta)+1)
dt=tibble::tibble(x,y)
rotacao=c(pi/4, pi/2, 3*pi/4, pi, 5*pi/4, 6*pi/4, 7*pi/4 )
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 = 0.75; xt4=xt*contracao; yt4=yt*contracao
dt4=tibble::tibble(xt4,yt4)
p=ggplot()+ coord_fixed()+theme_void()
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color='black',size=0.25)
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color='black',size=.25)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3.-5. Calcula aos valores x e y. Constrói banco de dados.

6. Define os ângulos de rotações e faz cópia dos valores de x e y.

7.-11. Calcula xt e yt utilizando a fórmula de rotação. Constrói o banco de dados.

12.-18. Define ângulos de rotação, comprimento do vetor xt e rotaciona xt por ângulos no vetor de rotação. Constrói banco de dados $dt1$.

19.-20. Define a razão da homotetia e calcula os respectivos $xt2$ e $yt2$. Constrói banco de dados $dt2$.

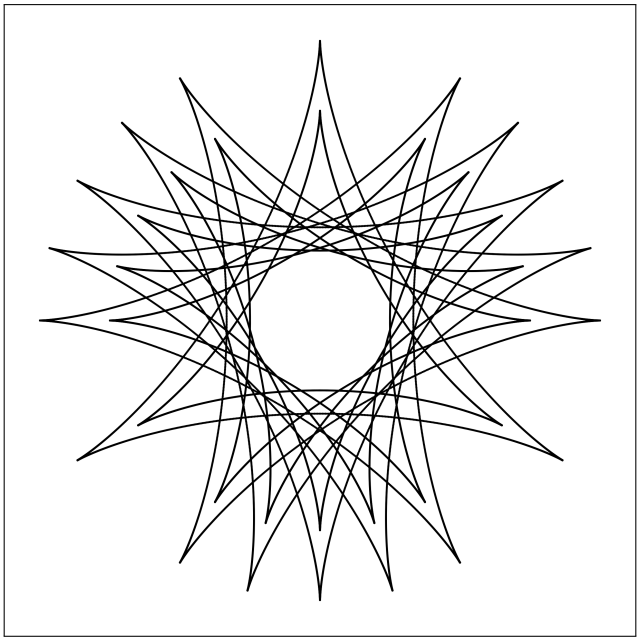
21.-25. Parâmetros gráficos.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

5.5 Deltoide

Neste caso, um deltoide é utilizado como base principal para as rotações e homotetias. A Figura 5.5 ilustra o resultado.

Figura 5.5: Composição de rotações e homotetias do deltoide.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=1500;  theta=seq(0,2*pi, length.out = n)
x=2*cos(theta)+cos(2*theta);  y=2*sin(theta)-sin(2*theta)
z=rep(0,n); dt=tibble::tibble(x,y,z)
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]))  }
z=c(dt$z,rep(c(pi/4,pi/2,3*pi/4,pi, 3*pi/2),c(n,n,n,n,n)))
dt=tibble::tibble(xt,yt,z)
contracao = 0.75;  xt4=xt*contracao; yt4=yt*contracao
dt4=tibble::tibble(xt4,yt4,z)
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=xt, y=yt), color='black',size=0.15)
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color='black',size=0.15)
p
```

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3. Calcula aos valores x e y.

4. Determina os valores z e constrói o banco de dados.

5. Define os ângulos de rotação.

6.-09 Faz cópia dos vetores x e y. Calcula xt e yt utilizando a fórmula de rotação.

10.-11 Define os valores z e o banco de dados.

12.-13. Define a razão da homotetia, calcula os respectivos valores. Define o banco de dados.

14.-17. Parâmetros gráficos.

De forma geral, o processo de construção pode ser resumido como segue:

- Construção do deltoide.
- Rotações sucessivas por ângulos $\theta = \frac{\pi}{4}, \frac{\pi}{2}, 3 \cdot \frac{\pi}{4}, \pi, 3 \cdot \frac{\pi}{2}$.
- Cálculo de x_t e y_t, com todos os valores em x e y rotacionados.
- Homotetia de razão $0,75$ para x_t, y_t.
- Composição da figura com os resultados gerados.

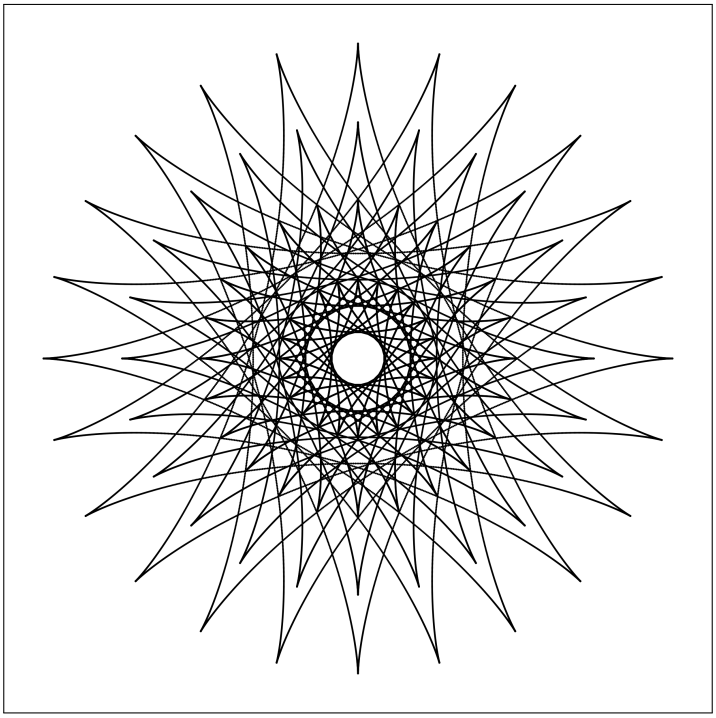
5.5.1 Efeito de sobreposição de rotações e homotetias

A Figura 5.6 mostra os resultados devido as modificações nas quantidades de rotações e homotetias.

- Construção do deltoide.
- Rotações sucessivas por ângulos múltiplos de $i \cdot \pi/4, i = 1, \cdots, 7$.
- Cálculo de x_t e y_t com todos os valores em x e y rotacionados.
- Homotetias de razão $0,25$, $0,5$ e $0,75$ para x_t, y_t.
- Composição da figura com os resultados gerados.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 5.6: Composição de rotações e homotetias do deltoide.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
n=1500
theta=seq(0,2*pi, length.out = n)
x=2*cos(theta)+cos(2*theta)
y=2*sin(theta)-sin(2*theta)
z=rep(0,n)
dt=tibble::tibble(x,y,z)
rotacao=pi/4*(1:7)
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])) }
xtt=NULL; ytt=NULL; red=c(0.25, 0.5, 0.75)
for(i in 1:length(red)){
provx=paste0("x",i); provy=paste0("y",i)
xtt=c(xtt, assign(provx, xt*red[i]))
ytt=c(ytt, assign(provy, yt*red[i])) }
dt=data.frame(x=c(xt, xtt), y=c(yt, ytt), z="deltóide")
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=x, y=y), color='black',size=0.05)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3.-4 Calcula aos valores x e y.

5.-6 Determina os valores z e constrói o banco de dados.

7. Define os ângulos de rotação.

8.-11 Faz cópia dos vetores x e y. Calcula xt e yt utilizando a fórmula de rotação.

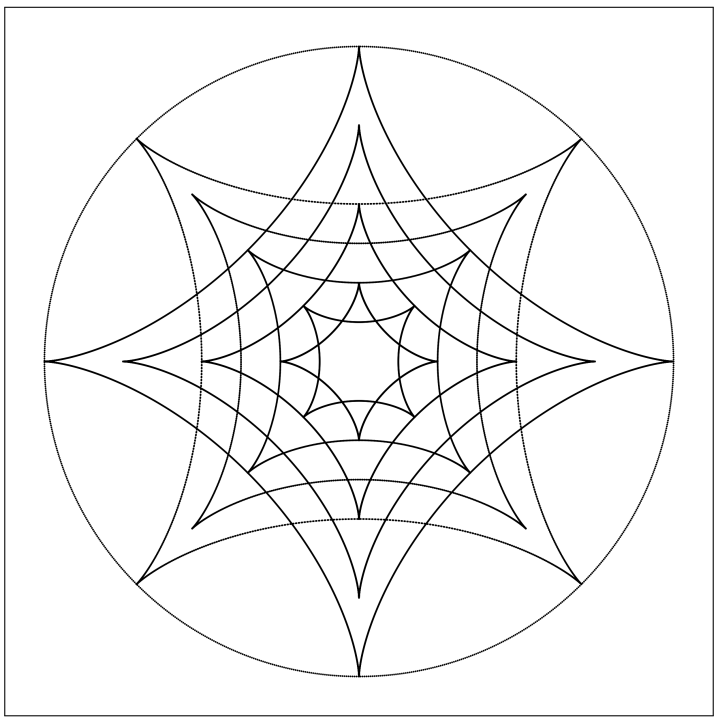
12.-17 Cálculos das homotetias.

18.-20. Parâmetros gráficos.

5.6 Astroide

Neste caso, um astroide é utilizado como base principal e, em seguida, rotações e homotetias são utilizadas. A Figura 5.7 ilustra o resultado.

Figura 5.7: Rotações e homotetias do astroide.



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=1500;   theta=seq(0,2*pi, length.out = n)
x=3*cos(theta)^3;      y=3*sin(theta)^3
z=rep(0,n);    dt=tibble::tibble(x,y,z)
rotacao=c(pi/4);     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])) }
z=c(dt$z,rep(rotacao,c(n,n)));

dt=tibble::tibble(xt,yt,z)
contracao = 0.25; xt2=xt*contracao;  yt2=yt*contracao
dt2=tibble::tibble(xt2,yt2,z)
contracao = 0.5;  xt3=xt*contracao;  yt3=yt*contracao

dt3=tibble::tibble(xt3,yt3,z)
contracao = 0.75;  xt4=xt*contracao; yt4=yt*contracao
dt4=tibble::tibble(xt4,yt4,z)

p= ggplot()+ coord_fixed()+ theme_void()
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color='black')
p = p+geom_point(data=dt2, aes(x=xt2, y=yt2), color='black')
p = p+geom_point(data=dt3, aes(x=xt3, y=yt3), color='black')
p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color='black')
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3. Calcula aos valores x e y.

4. Determina os valores z e constrói o banco de dados.

5. Define os ângulos de rotação. Faz cópia dos vetores x e y.

6.-9 Calcula xt e yt utilizando a fórmula de rotação.

10. Define os valores z e o banco de dados.

11.-15. Define a razão da homotetia e calcula os respectivos valores.

17.-21. Parâmetros gráficos.

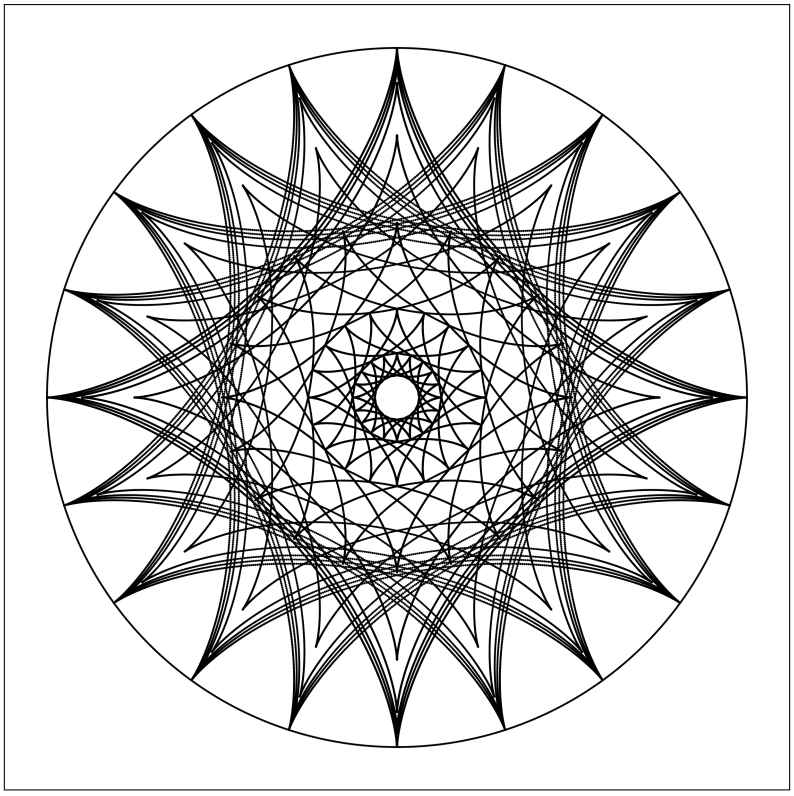
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

5.6.1 Efeitos de sobreposição de rotações e homotetias

Neste caso, as rotações e homotetias são escolhidas de forma distinta daquela da Figura anterior. O procedimento é idêntico ao caso anterior.

- Construção do astroide.
- Rotações sucessivas por ângulos $\theta = i \cdot \frac{\pi}{5}, i = 1, \cdots, 5$.
- Cálculo de x_t e y_t com todos os valores em x e y rotacionados.
- Homotetias de razão $r = 0,125; 0,25; 0,5; 0,75; 0,9; 0,95; 0,975; 1$ para x_t, y_t.
- Composição da figura com os resultados gerados.

Figura 5.8: Composição de rotações e homotetias do astroide.



Fonte: Os autores.

Note que as rotações são em múltiplos de $\pi/5$, mas as homotetias não são regularmente espaçadas no intervalo $[0, 1]$, ocasionando maior densidade de linhas próximos ao astroide original ou rotações.

O código completo é apresentado a seguir:

```
require(ggplo2)
n=1500; theta=seq(0,2*pi, length.out = n)
x=cos(theta)^3; y=sin(theta)^3
z=rep(0,n); dt=tibble::tibble(x,y,z)
rotacao=pi/5*(1:5); 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]))}
xtt=NULL; ytt=NULL; red=c(.125,0.25, 0.5, 0.75,.9,.95,.975,1.)
for(i in 1:length(red)){
provx=paste0("x",i); provy=paste0("y",i)
xtt=c(xtt, assign(provx, xt*red[i]))
ytt=c(ytt, assign(provy, yt*red[i])) }
dt=data.frame(x=c(xt, xtt), y=c(yt, ytt), z="astroide")
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=x, y=y), color='black',size=0.05)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$.

3. Calcula aos valores x e y.

4. Determina os valores z e constrói o banco de dados.

5. Define os ângulos de rotação. Faz cópia dos vetores x e y.

6.-8 Calcula xt e yt utilizando a fórmula de rotação.

7. Definições do vetor para as homotetias.

10.-14. Executa os cálculos de homotetias. Constrói banco de dados.

15.-17. Parâmetros gráficos.

5.7 Circunferências: translações e rotações

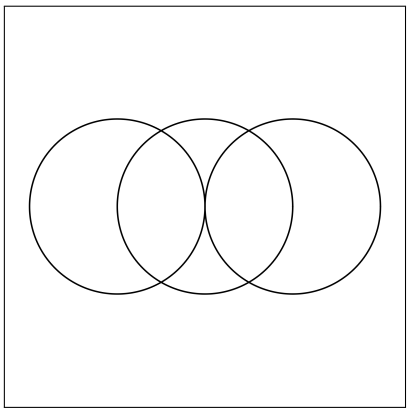
O processo de construção consiste em deslocar um circunferência central para esquerda e direita, conforme mostrado na Figura 5.9. A figura obtida é então rotacionada em torno da origem. O procedimento inicial consiste em construir a circunferência com centro em $(0, 0)$;

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

realizar translações sucessivas e horizontais de distâncias r e $2r$. A seguir estão o código e a respectiva Figura 5.9.

Figura 5.9: Translação.

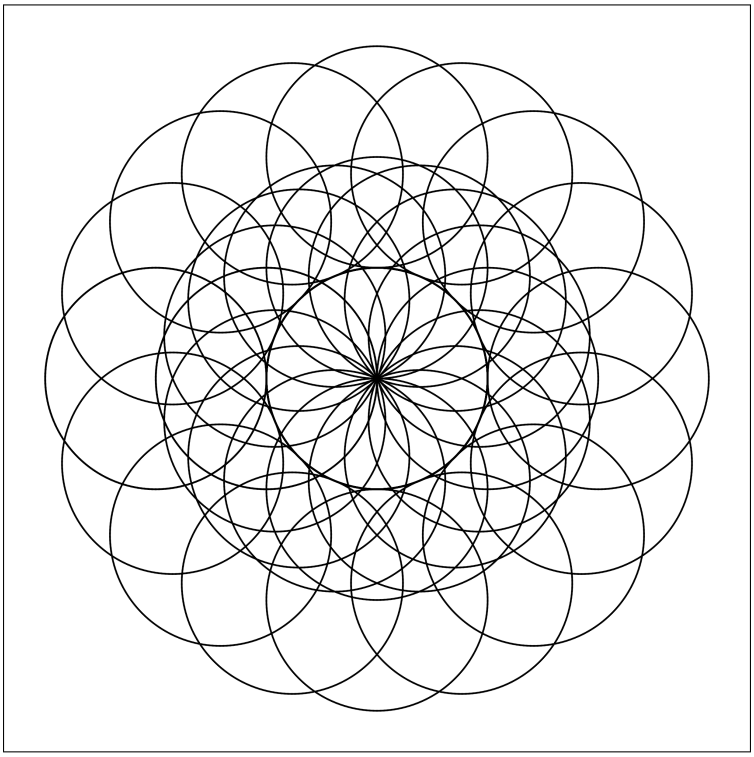
```
require(ggplot2); n=500
theta=seq(0,2*pi, length.out = n)
raio=1; x=raio*cos(tetha); y=raio*sin(tetha)
xt=c(x,x-raio, x-2*raio); yt=c(y,y,y)
dt=tibble::tibble(xt,yt)
p=ggplot()+coord_fixed()+theme_void()
p=p+ geom_point(data=dt, aes(x=xt, y=yt),
color='black',size=.5)
p
```



Fonte: Os autores.

A composição de rotações dos pontos armazenados na tabela de dados (dt), que consiste de valores múltiplos de $\pi/8$ dá origem a Figura 5.10.

Figura 5.10: Rotações das três circunferências da Figura 5.9.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n); raio=1
x=raio*cos(theta); y=raio*sin(theta)
xt=c(x,x-raio, x-2*raio); yt=c(y,y,y);  dt=tibble::tibble(xt,yt)
rotacao=pi/8*1:16; n=length(xt); 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]))}
dt1=tibble::tibble(xt1,yt1)
p=ggplot()+coord_fixed()+ theme_void()
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color='black',size=0.5)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos e calcula a sequência de pontos `theta` no intervalo $[0, 2\pi]$ e o raio.

3. Calcula aos valores x e y.

4. Calcula os vetores xt dos deslocamentos e o vetor y respectivo. Constrói o banco de dados.

5. Define os ângulos de rotação e o comprimento do vetor. Faz cópia dos vetores xt e yt.

6.-9 Calcula $xt1$ e $yt1$ utilizando a fórmula de rotação. Constrói o banco de dados.

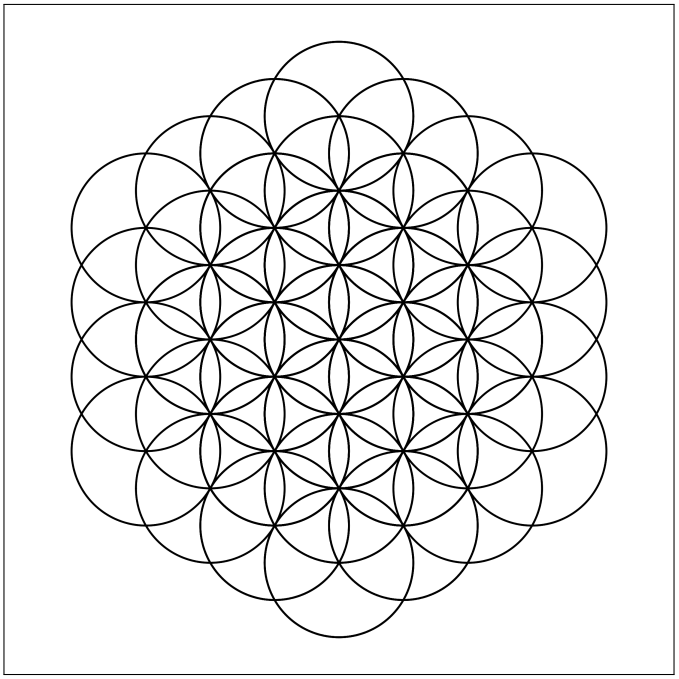
10. Parâmetros gráficos.

5.7.1 Rotações de várias circunferências

Neste caso, translações e rotações são utilizadas na composição da Figura. O processo de construção pode ser resumido como segue:

- Construção da circunferência central e translação para obtenção de 5 circunferências verticais.
- Rotação da construção por ângulos $i \cdot \pi/k, i = 1, 2, \cdots, 2k$ com $k = 3$.

Figura 5.11: Rotações de várias circunferências transladadas.



Fonte: Os autores.

O código completo é dado a seguir:

```
require(ggplot2)
n=500; raio=1; t=seq(0,2*pi, length.out = n)
x=raio*cos(t); y=raio*sin(t);
xt=c(x);  yt=c(y+1*raio);
k=3;  rotacao=pi/k*(1:(2*k))
xtt=c();ytt=c(); n=length(xt)
for(i in 1:length(rotacao)){
xtt=c(xtt,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
ytt=c(ytt,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))}
dt=data.frame(x=c(xtt),y=c(ytt),z="circulo-flor")
x=dt$x; y=dt$y
dt = data.frame(x = c(x,x,x,x,x), y=c(y-2,y-1,y, y+1,y+2))
k=3;rotacao=pi/k*(1:(2*k))
xt=dt$x; yt=dt$y;
xtt=xt; ytt=yt; n=length(xt)
for(i in 1:length(rotacao)){
xtt=c(xtt,xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
ytt=c(ytt,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))}
dt1=data.frame(x=c(xtt),y=c(ytt),z="circulo-flor")
p=ggplot()+coord_fixed()+    theme_void()
size=0.15
p=p+ geom_point(data=dt1, aes(x=x, y=y), color='black',size=size)
p
```

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

24

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos, define o raio e calcula a sequência de pontos `t` no intervalo $[0, 2\pi]$.

3. Calcula os valores x e y.

4. Calcula os vetores xt e o vetor deslocado yt.

5. Define o parâmetro k. Define os ângulos de rotação.

6. Faz cópia dos vetores xtt e ytt. Define o comprimento do vetor xt.

7.-10. Calcula xtt e ytt utilizando a fórmula de rotação. Constrói o banco de dados. Constrói o banco de dados.

11.-12. Redefinição e construção de banco de dados com circunferências transladados na direção y.

13.-18. Similar a 4-12 para rotacionar da figura vertical formada em 4-12.

19.-23. Parâmetros gráficos.

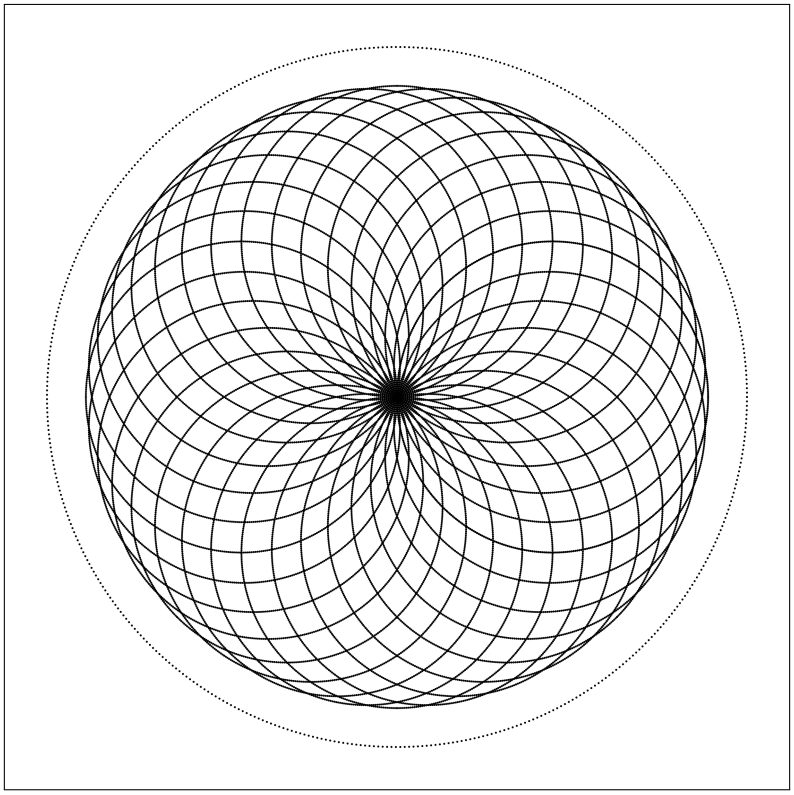
5.8 Rotação de circunferência em torno da origem

A Figura 5.12 foi gerada por meio de circunferências rotacionadas por um ângulo específico. O procedimento de construção consiste:

- Construir uma circunferência de centro na origem e raio R deslocada para a direita em uma unidade de raio.
- Realizar rotações sucessivas em torno da origem.
- Composição da figura.
- Construção de uma circunferência em torno da composição anterior.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 5.12: Rotação de circunferências em torno da origem.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500;  theta=seq(0,2*pi, length.out = n); raio=1
x=raio*cos(theta);   y=raio*sin(theta)
xt=c(x+1.*raio);  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,xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))}
dt1=tibble::tibble(xt1,yt1)
theta=seq(0,2*pi, length.out = n); raio=2.25
x=raio*cos(theta); y=raio*sin(theta)
xt=c(x); yt=c(y)
dt=tibble::tibble(xt,yt)
p=ggplot()+coord_fixed()+    theme_void()
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color='black',size=.05)
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color='black',size=.05)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos, define o raio e calcula a sequência de pontos `t` no intervalo $[0, 2\pi]$.

3. Calcula os valores x e y.

4. Calcula os vetores xt deslocado e o vetor yt.

5. Define o parâmetro $n1$. Define os ângulos de rotação.

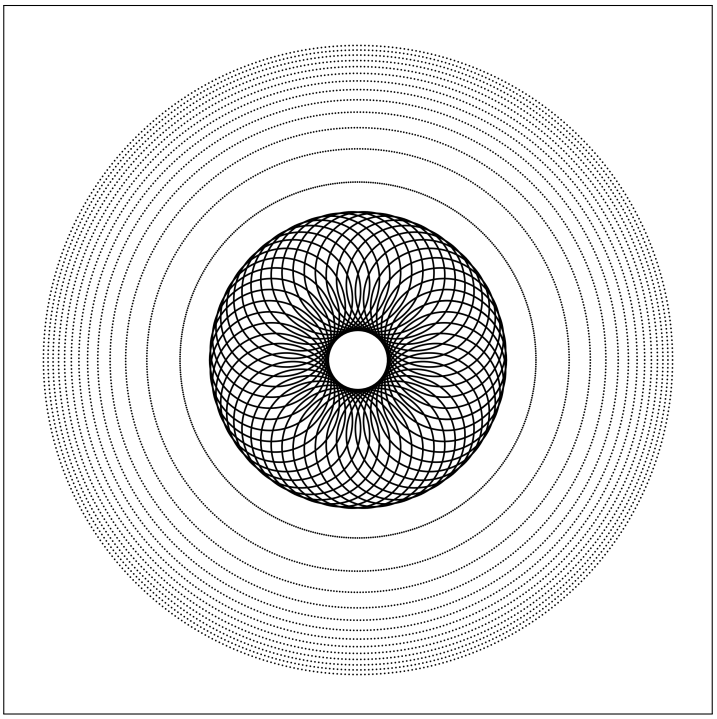
6.-10. Rotaciona por ângulos dados e constrói banco de dados.

11.-14. Constrói circunferência de raio maior e banco de dados.

15.-18. Parâmetros gráficos.

A alteração na quantidade de rotações n_1 e inserção de uma sequência de circunferências concêntricas pode adicionar elementos interessantes tais como apresentados a seguir. Ver Figura 5.12 e que a figura anterior também foi alterada de modo que o raio é cerca de 1.25, mas o centro de rotação é idêntico.

Figura 5.13: Composição de sequência de circunferências concêntricas na Figura 5.12.



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

A Figura 5.13 difere da Figura 5.12 pela presença de uma sequência de circunferências concêntricas com raios variando em escala logarítmica.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n); raio=1
x=raio*cos(theta); y=raio*sin(theta)
xt=c(x+1.5*raio); yt=c(y)
n=25; rotacao=pi/n*1:(2*n); n=length(xt)
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]))}
dt1=tibble::tibble(xt1,yt1)
theta=seq(0,2*pi, length.out = n); raio=3.0;  n=length(xt)
x=raio*cos(theta); y=raio*sin(theta)
xt2=c(); yt2=c()
for(i in log(seq(1,10,.75))){
  raio=3+i; x=raio*cos(theta); y=raio*sin(theta)
  xt2=c(xt2,x[1:n]*1);  yt2=c(yt2,y[1:n]*1) }
dt2=tibble::tibble(xt2,yt2)
p=ggplot()+coord_fixed()+    theme_void()
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color='black',size=.015)
p=p+ geom_point(data=dt2, aes(x=xt2, y=yt2), color='black',size=.015)
p
```

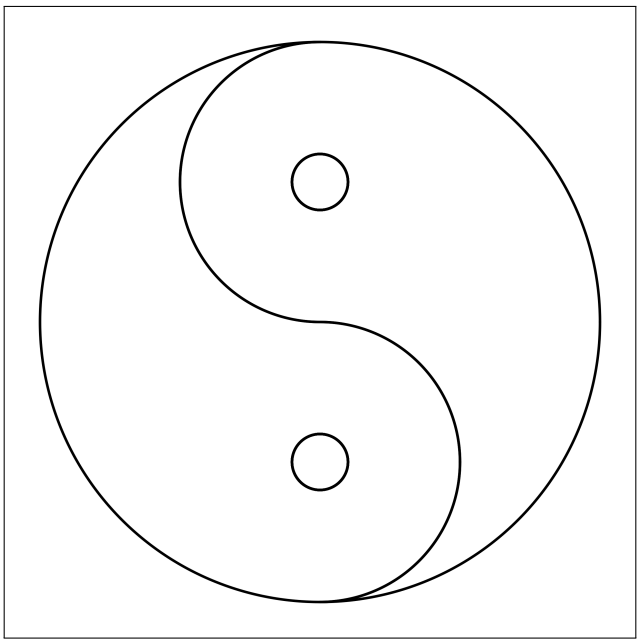
5.9 Composição de circunferências

A Figura 5.14 é facilmente reconhecida como o símbolo Ying-Yang. O procedimento construtivo é o seguinte:

- Definir uma circunferência de raio r com centro em $O(0,0)$.
- Traçar um diâmetro e calcular os pontos médios das de cada um dos raios sobre o diâmetro.
- Determinar 2 novas circunferências de raios $r/4$.
- Determinar os pontos de interesse e excluir regiões.
- Compor a figura.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 5.14: Composição de circunferências.



Fonte: Os autores.

O código é dado a seguir:

```
require(ggplot2)
n=1600; raio=4; t=seq(0,2*pi, length.out = n)
x=raio*cos(t); y=raio*sin(t)
xt=c(x, 0.1*x,  0.1*x,   0.5*x[(n/4):(3*n/4)],   0.5*x[c(1:(n/4),(3*n/4):n)] )
yt=c(y, 0.1*y-2,0.1*y+2, 0.5*y[(n/4):(3*n/4)]+2, 0.5*y[c(1:(n/4),(3*n/4):n)]-2)
dt=data.frame(xt,yt,z="circulo")
p= ggplot()+coord_fixed()+theme_void()
p=p+  geom_point(data=dt, aes(x=xt, y=yt), color='black',size=0.5)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos, o raio da circunferência maior e calcula a sequência de pontos `t` no intervalo $[0, 2\pi]$.

3. Calcula os vetores x e y da circunferência maior.

4.-5. Constrói vector de pontos da circunferência maior, das circunferências menores localizadas sobre o eixo vertical e os pontos das semi-circunferências.

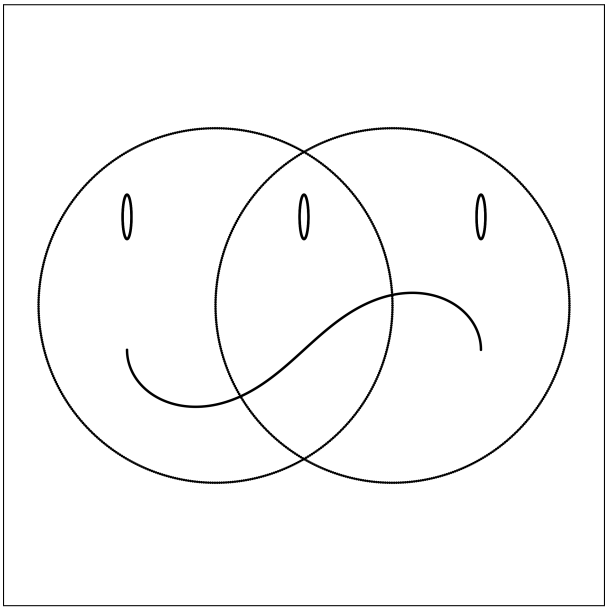
6.-9. Constrói banco de dados. Parâmetros gráficos.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

5.10 Composição de circunferências, elipses e lemniscata

A Figura 5.15, chamada Mandala Sentimentos, é uma composição de circunferências, elipses e uma parte da lemniscata de Bernoulli.

Figura 5.15: Composição de curvas planas.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n);  raio=2
x1=raio*cos(theta);  y1=raio*sin(theta)
deslocamento = c(-1,1) ;   x=NULL;y=NULL
for(i in 1:length(deslocamento)){
x=c(x,x1+deslocamento[i]);    y=c(y,y1)}
td1 = data.frame(x,y,z="sentimento")
x1=c(0.05*cos(theta),0.05*cos(theta)-2,0.05*cos(theta)+2)
y1=c(0.25*sin(theta)+1,0.25*sin(theta)+1,0.25*sin(theta)+1)
td2 = data.frame(x=x1,y=y1,z="sentimento")
t=seq(-pi/2,pi/2, length.out = n)
a=2; x=a*sin(t)/(1+cos(t)^2)
y=a/1.1*cos(t)*sin(t)/(1+cos(t)^2)-0.5
td3 = data.frame(x,y,z="sentimento")
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=td1, aes(x=x, y=y), color='black',size=0.5)
p=p+geom_point(data=td2, aes(x=x, y=y), color='black',size=0.5)
p=p+ geom_point(data=td3, aes(x=x, y=y), color='black',size=0.5)
p
```

O detalhamento do código é feito a seguir:

1.-2 Carrega o pacote *ggplot2*. Define a quantidade de pontos n a ser utilizada para os cálculos, calcula a sequência de pontos `t` no intervalo $[0, 2\pi]$ e define o raio da circunferência.

3.-4 Define as vetores x e y para uma circunferência com centro na origem. Define os deslocamentos e os vetores x e y.

5.-7. Calcula os pontos das circunferências deslocadas. Constrói banco de dados.

8.-10. Define as elipses dos olhos.

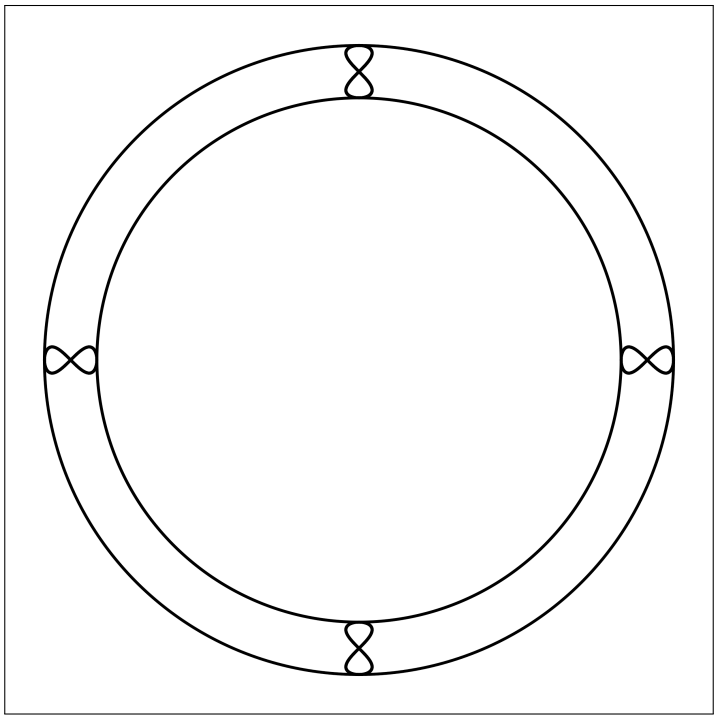
11.-14. Construção da leminscata parcial.

15.-19. Parâmetros gráficos.

5.11 Circunferências e lemniscatas

A Figura 5.16, chamada Mandala Escudo, é uma composição de circunferências e lemniscatas com homotetias e rotações.

Figura 5.16: Mandala escudo



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=1500; theta=seq(0,2*pi, length.out = n)
raio1=1; raio2=1
x=raio1*cos(theta);  y=raio2*sin(theta)
dt=tibble::tibble(x,y)
reducao=0.2;
x1=sin(theta)*cos(theta)*reducao/2
y1=sin(theta)*reducao/2+(1+reducao/2)
dt1=tibble::tibble(x1,y1)
raio1=1.2; raio2=1.2
x2=raio1*cos(theta); y2=raio2*sin(theta)
dt2=tibble::tibble(x2,y2)
size=0.5
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=x, y=y), color='black',size=size)
p=p+geom_point(data=dt1, aes(x=x1, y=y1), color='black',size=size)+
geom_point(data=dt2, aes(x=x2, y=y2), color='black',size=size)
rotacao=c(pi/2,pi, 3*pi/2)
xt=x1; yt=y1
for(i in 1:length(rotacao)){
xt=c(xt,x1[1:n]*cos(rotacao[i])-y1[1:n]*sin(rotacao[i]))
yt=c(yt,x1[1:n]*sin(rotacao[i])+y1[1:n]*cos(rotacao[i]))  }
dt1=tibble::tibble(xt,yt)
p=p+geom_point(data=dt1, aes(x=xt, y=yt), color='black',size=size)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos, calcula a sequência de pontos `t` no intervalo $[0, 2\pi]$ e define os raios das circunferências.

3.-5. Define as vetores x e y para uma circunferência com centro na origem. Define o banco de dados.

6.-9. Define a lemniscata e fator de homotetia. Constrói banco de dados.

10.-12. Idêntico a circunferência maior.

13.-17. Parâmetros gráficos.

18.-23. Rotação da leminscata por ângulos definidos.

24.-25. Parâmetros gráficos.

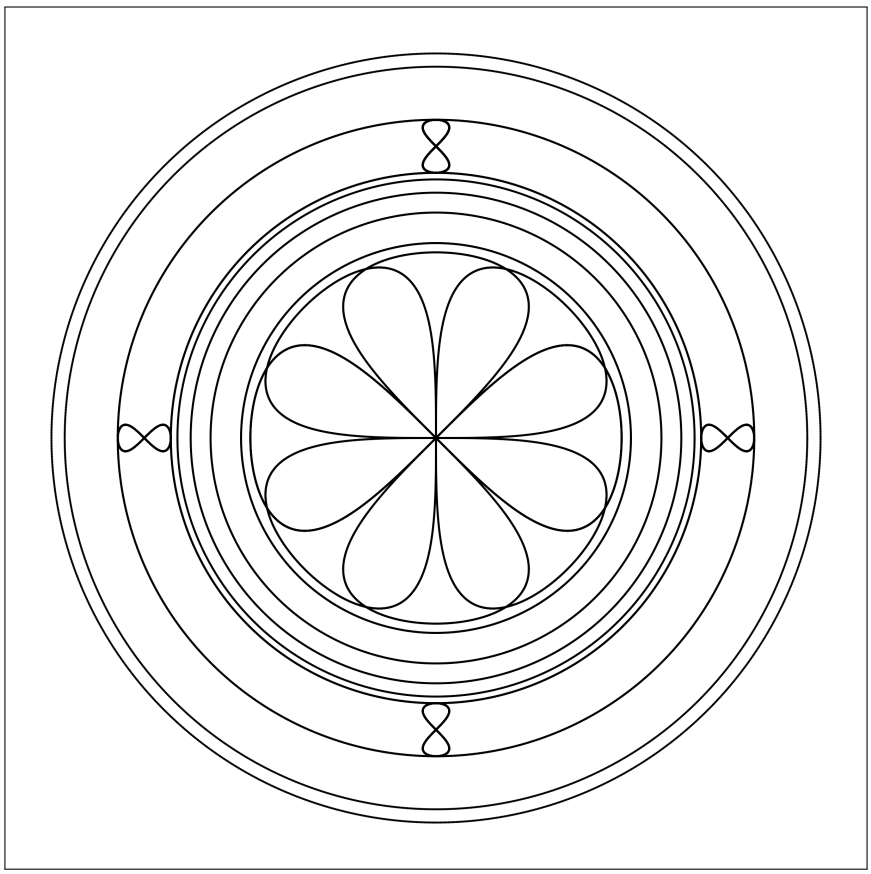
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

5.11.1 Circunferências e lemniscatas

A Figura 5.16, chamada Mandala Escudo Modificado, insere uma combinação de circunferências de raios distintos com rotações e homotetias da lemniscata de Bernoulli. O procedimento é o seguinte:

- Construção das circunferências mais externas e com centro na origem.
- Construção da homotetia da lemniscata menor e rotações.
- Construção da sequência de circunferências com centro na origem.
- Construção da lemniscata centrada na origem e rotações.
- Composição da figura.

Figura 5.17: Mandala escudo



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

O código completo é mostrado a seguir:

```
n=1500;  theta=seq(0,2*pi, length.out = n)
raio1=1; raio2=1
x=raio1*cos(theta); y=raio2*sin(theta)
dt=tibble::tibble(x,y)
reducao=0.2
x1=sin(theta)*cos(theta)*reducao/2; y1=sin(theta)*reducao/2+(1+reducao/2)
dt1=tibble::tibble(x1,y1)
raio1=1.2; raio2=1.2
x2=raio1*cos(theta); y2=raio2*sin(theta)
dt2=tibble::tibble(x2,y2)

size=0.05
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=x, y=y), color='black',size=size)
p=p+geom_point(data=dt1, aes(x=x1, y=y1), color='black',size=size)+
geom_point(data=dt2, aes(x=x2, y=y2), color='black',size=size)#+

rotacao=c(pi/2,pi, 3*pi/2); xt=x1; yt=y1
for(i in 1:length(rotacao)){
xt=c(xt,x1[1:n]*cos(rotacao[i])-y1[1:n]*sin(rotacao[i]))
yt=c(yt,x1[1:n]*sin(rotacao[i])+y1[1:n]*cos(rotacao[i]))  }
dt1=tibble::tibble(xt,yt)
p=p+geom_point(data=dt1, aes(x=xt, y=yt), color='black',size=.015)
reducao=.7;
x3=sin(theta)*cos(theta)*reducao; y3=sin(theta)*reducao
dt3=tibble::tibble(x3,y3)
rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
xt3=c(); yt3=c()
for(i in 1:length(rotacao)){
xt3=c(xt3,x3[1:n]*cos(rotacao[i])-y3[1:n]*sin(rotacao[i]))
yt3=c(yt3,x3[1:n]*sin(rotacao[i])+y3[1:n]*cos(rotacao[i]))}
dt3=tibble::tibble(xt3,yt3)
p=p+geom_point(data=dt3, aes(x=xt3, y=yt3), color='black',size=.015)
xt4=c(); yt4=c()
for(i in c(0.7,0.75,0.275,0.225,0.15,0.035)){
  raio=.7+i
  x=raio*cos(theta); y=raio*sin(theta)
  xt4=c(xt4,x[1:n]*1); yt4=c(yt4,y[1:n]*1)}
dt4=tibble::tibble(xt4,yt4)
p=p+geom_point(data=dt4, aes(x=xt4, y=yt4), color='black',size=.015)
p
```

O detalhamento do código é uma combinação de elementos apresentados anteriormente.

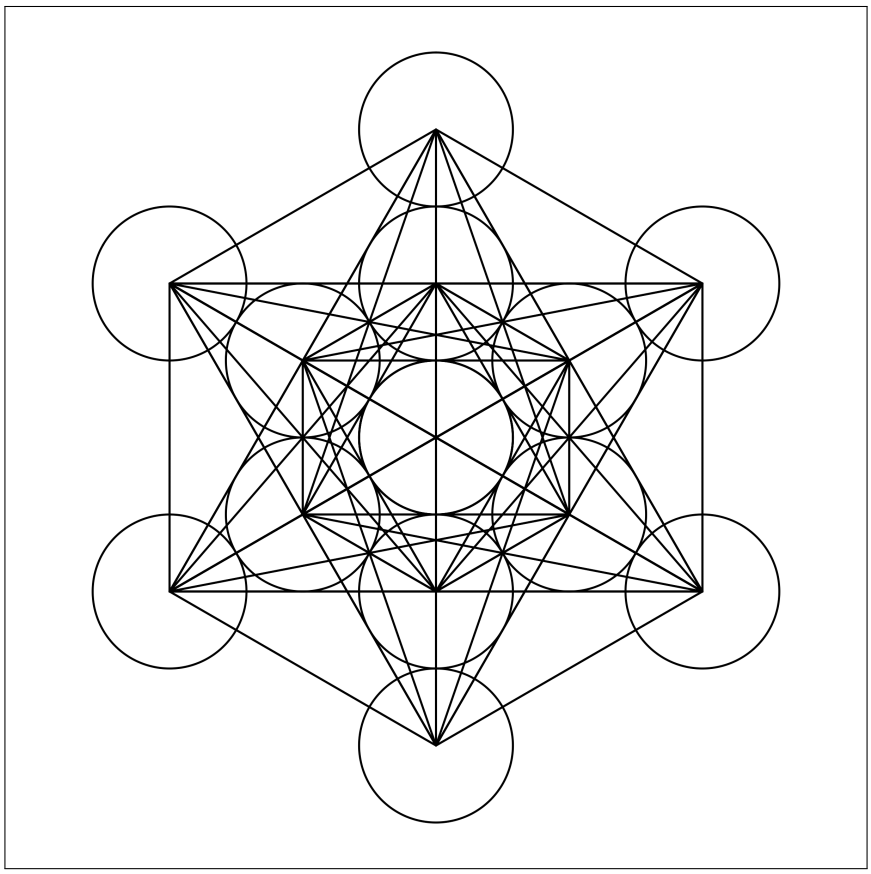
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

5.12 Circunferências e segmentos

Nesta parte, o chamado cubo de Metatron ilustra a utilização de programação para a solução de um problema de combinar todos os centros de um conjunto específico de circunferências por meio de segmentos de retas. De forma mais sistemática é uma figura geométrica composta de 13 circunferências, nos quais todos os centros são interconectados por segmentos de retas. A Figura 5.18 ilustra a construção cujo o procedimento construtivo é mostrado a seguir:

- Construção da circunferência c_1 de centro na origem e raio r. Construção de outras quatro circunferências transladadas de forma simétrica em relação ao centro de c_1.
- Rotação do conjunto de circunferências do item anterior em torno da origem de modo que os centros das circunferências externas formem um hexágono.
- Construção dos segmentos de retas ligando os centros das circunferências.

Figura 5.18: Translações, rotações e segmentos de retas



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n);  raio=1
deslocamento = c(-4,-2,0,2,4)
x1=seq(-4,4,length.out = n); y1=rep(0,length(x1))
for(i in 1:length(deslocamento)){
x1=c(x1,raio*cos(theta)+deslocamento[i])
y1=c(y1,raio*sin(theta))}
rotacao=pi/6*seq(1,5,2);  n=length(x1)
xt1=NULL; yt1=NULL
for(i in 1:length(rotacao)){
xt1=c(xt1,x1[1:n]*cos(rotacao[i])-y1[1:n]*sin(rotacao[i]))
yt1=c(yt1,x1[1:n]*sin(rotacao[i])+y1[1:n]*cos(rotacao[i]))}
xd0=deslocamento;  yd0=rep(0,length(deslocamento))
xd=NULL; yd=NULL
for(i in 1:length(rotacao)){
xd=c(xd,xd0[1:length(deslocamento)]*cos(rotacao[i])-
yd0[1:length(deslocamento)]*sin(rotacao[i]))
yd=c(yd,xd0[1:length(deslocamento)]*sin(rotacao[i])+
yd0[1:length(deslocamento)]*cos(rotacao[i])) }
td = data.frame(xd,yd)
rep=500;     x=NULL;     y=NULL;    n=(length(xd)-1)
for(i in 1:n){
  for(j in i:n){
    x=c(x,seq(xd[i],xd[j+1],length.out=rep))
    y=c(y,seq(yd[i],yd[j+1],length.out=rep))} }
df1=tibble::tibble(x,y,z="metatron")
dt1=tibble::tibble(xt1,yt1,z="metatron")
p=ggplot()+coord_fixed()+ theme_void();size=0.015
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color='black',size=size)
p = p+geom_point(aes(x = x, y), data = df1,size=size)
p
```

O detalhamento do código é feito a seguir:

1. Carrega o pacote *ggplot2*.

2. Define a quantidade de pontos n a ser utilizada para os cálculos, calcula a sequência de pontos `t` no intervalo $[0, 2\pi]$ e define o raio das circunferências.

3. Define os deslocamentos simétricos em torno da origem.

4. Calcula os vetores $x1$ e $y1$ para construir as circunferências.

5.-7. Calcula os pontos sobre as circunferências com centro paralelo ao eixo x.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

8.-9. Define os ângulos de rotação e o comprimento do vetor para rotações. Cria vetores vazios.

10.-12. Faz as rotações das circunferências.

13.-20. Define os centros das circunferências rotacionadas por meio de rotações dos centros das circunferências iniciais.

21.-27. Constrói os segmentos de retas ligando os centros das circunferências rotacionadas.

28.-31. Parâmetros gráficos.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Capítulo 6

Cores em R

Quando pensamos em cores numa visualização, temos os seguintes aspectos a considerar quanto a aplicação das cores, podemos colorir o plano de fundo, ou os pontos que formam nosso desenho. Definir o esquema de cores para apresentação pode ser desafiador, pois um esquema de cores adequado pode ressaltar aspectos relevantes da figura. Aqui, as cores são escolhidas de forma a focar na apresentação dos aspectos computacionais, ou seja, a respectiva utilização para a visualização da figura. Uma discussão geral sobre cores no R pode ser encontrado, por exemplo, em [6].

O R utiliza o formato hexadecimal de representação de cores, o qual é um sistema de base 16 utilizada para descrever cada uma das cores. Cada um dos caracteres possui um dos símbolos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Por exemplo, $\#FF0000$ representa vermelho (*red*), #000000 representa preto (*black*) e $\#FFFFFF$ representa branco. O modelo RGB utiliza escalas de vermelho (*Red*), verde (*Green*) e azul (*Blue*). O modelo HSV utiliza matiz (*hue*), saturação (*saturation*) e valor (*value*) no intervalo de $[0, 1]$. O modelo HCL utiliza matiz (*hue*), croma (*croma*) e luminância (*luminance*). A matiz varia no intervalo de $0 - 360$, sendo vermelho (0), verde (120), azul (240). Maiores detalhes podem ser encontrados em [6], o qual foi baseado em [36].

Detalhes sobre elementos gráficos a serem coloridos com o pacote *ggplot2* e discussões sobre paletas de cores e seus tipos podem ser vistas em [1, p. 188-193]. Pacotes especializados podem ser utilizados. Exemplos específicos, que não esgotam as possibilidades são: **grDevices**, [25], **colorRamps**, [8], **colorspace**, [37],[36], [26].

No entanto, a utilização de cores no R pode ser feita de forma mais direta. O R traz uma lista de cores pré-configurada que pode ser acessada diretamente por meio do nome, além do respectivo esquema hexadecimal.

Os nomes de todas as cores que podem ser obtidas diretamente por meio da função $colors()$, a qual retorna uma lista com 657 possibilidades. A seguir estão os nomes das quarenta primeiras cores e o respectivo código em R utilizado.

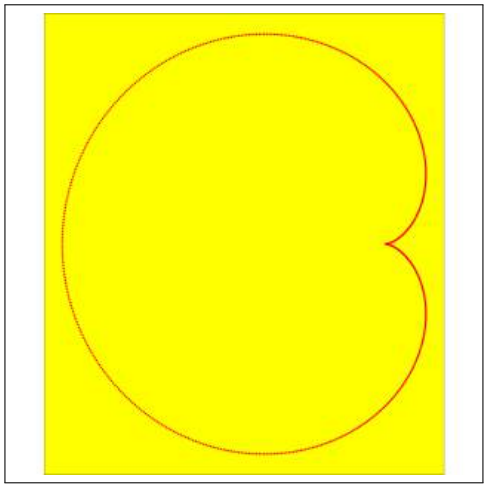
```
x=c(colors()); matrix(x[1:20],nrow = 5)
```

```
      [,1]          [,2]           [,3]           [,4]
 [1,] "white"       "aquamarine3"  "bisque2"      "blueviolet"
 [2,] "aliceblue"   "aquamarine4"  "bisque3"      "brown"
 [3,] "antiquewhite" "azure"       "bisque4"      "brown1"
 [4,] "antiquewhite1" "azure1"     "black"        "brown2"
 [5,] "antiquewhite2" "azure2"     "blanchedalmond" "brown3"
 [6,] "antiquewhite3" "azure3"     "blue"         "brown4"
 [7,] "antiquewhite4" "azure4"     "blue1"        "burlywood"
 [8,] "aquamarine"  "beige"       "blue2"        "burlywood1"
 [9,] "aquamarine1" "bisque"      "blue3"        "burlywood2"
[10,] "aquamarine2" "bisque1"     "blue4"        "burlywood3"
```

Estas cores podem ser utilizadas diretamente no R para alterar a coloração dos pontos do gráfico ou o plano de fundo. Por exemplo, a Figura 6.1 mostra um cardioide em cor vermelha e com plano de fundo amarelo. O código é similar àquele apresentado na seção e reproduzido a seguir, mas a descrição detalhada é omitida.

```
require(ggplot2)
n=700
theta=seq(0,2*pi, length.out = n)
raio=1
x=(1-cos(theta))*cos(theta)*raio
y=(1-cos(theta))*sin(theta)*raio
dt=tibble::tibble(x,y)
size=0.5
p = ggplot()+coord_fixed()+theme_void()
p = p+geom_point(data=dt, aes(x=x, y=y),
color='red',size=size)
p=p+theme(panel.background =
element_rect(fill = "yellow")  )
p
```

Figura 6.1: Fundo Amarelo.

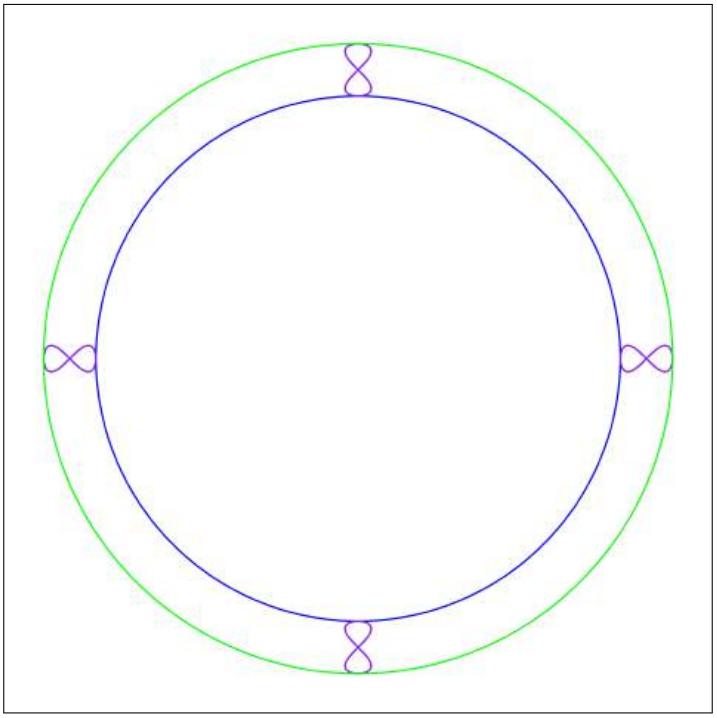


Fonte: Os autores.

A Figura 6.2 mostra um exemplo de utilização das cores azul, violeta azulado e verde.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.2: Composição com diferentes cores.



Fonte: Os autores.

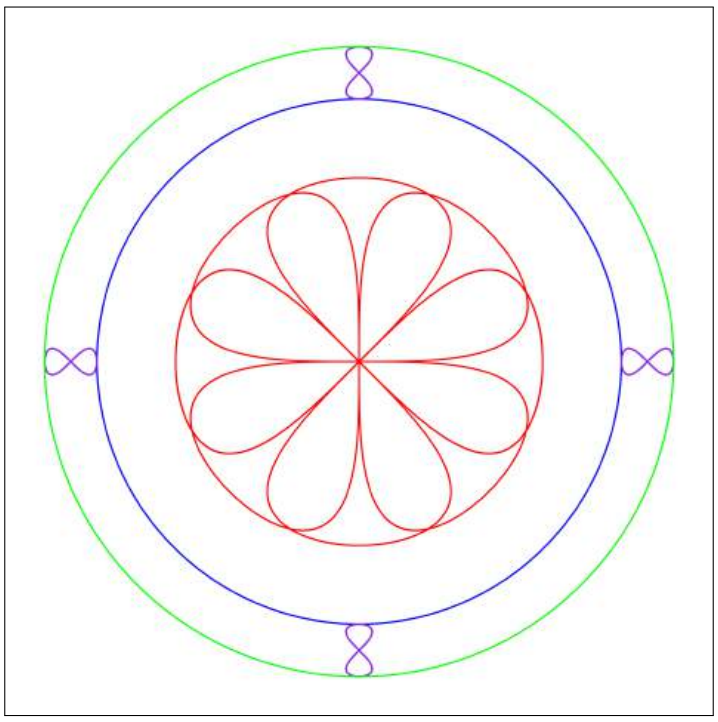
O código completo é apresentado a seguir:

```
require(ggplot2)
n=1500;  theta=seq(0,2*pi, length.out = n)
raio1=1; raio2=1
x=raio1*cos(theta); y=raio2*sin(theta)
dt=tibble::tibble(x,y)
reducao=0.2
x1=sin(theta)*cos(theta)*reducao/2; y1=sin(theta)*reducao/2+(1+reducao/2)
rotacao=c(pi/2,pi, 3*pi/2); xt=x1; yt=y1
for(i in 1:length(rotacao)){
  xt=c(xt,x1[1:n]*cos(rotacao[i])-y1[1:n]*sin(rotacao[i]))
  yt=c(yt,x1[1:n]*sin(rotacao[i])+y1[1:n]*cos(rotacao[i]))  }
dt1=tibble::tibble(xt,yt)
raio1=1.2; raio2=1.2
x2=raio1*cos(theta); y2=raio2*sin(theta)
dt2=tibble::tibble(x2,y2)
size=0.05
p=ggplot()+coord_fixed()+theme_void()
p=p+geom_point(data=dt, aes(x=x, y=y), color='blue',size=size)
p=p+geom_point(data=dt1, aes(x=xt, y=yt), color='blueviolet',size=size)+
  geom_point(data=dt2, aes(x=x2, y=y2), color='green',size=size)
p
```

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Analogamente, ao caso anterior, a descrição detalhada é omitida, pois apenas o parâmetro *color* das linhas 20 a 22 foi modificado. Figuras com mais camadas podem utilizar a definição manual das cores definidas de forma similar. A Figura 6.3 é um exemplo.

Figura 6.3: Composição com diferentes cores.



Fonte: Os autores.

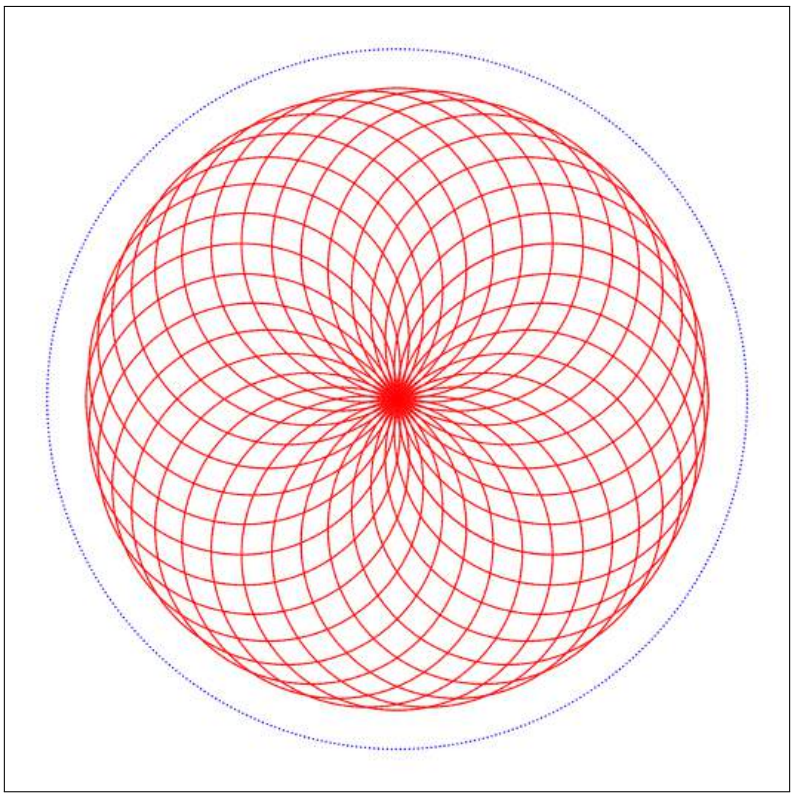
Deve ser notado que todos os pontos, obtidos da rotação da lemniscata, possuem mesma cor devido a forma de construção do vetor $(xt3, yt3)$.

```
1   reducao=.7;
2   x3=sin(theta)*cos(theta)*reducao; y3=sin(theta)*reducao
3   dt3=tibble::tibble(x3,y3)
4   rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2)
5   xt3=c(); yt3=c()
6   for(i in 1:length(rotacao)){
7     xt3=c(xt3,x3[1:n]*cos(rotacao[i])-y3[1:n]*sin(rotacao[i]))
8     yt3=c(yt3,x3[1:n]*sin(rotacao[i])+y3[1:n]*cos(rotacao[i]))}
9   dt3=tibble::tibble(xt3,yt3)
10  p=p+geom_point(data=dt3, aes(x=xt3, y=yt3), color='red',size=.015)
11  p
```

Analogamente à região central da figura anterior, a Figura 6.4 mostra o resultado de uma cor única aplicada às rotações, em torno da origem, de circunferência transladada.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.4: Composição de rotações com cores idênticas.



Fonte: Os autores.

O código completo é mostrado a seguir:

```
require(ggplot2)
n=500; theta=seq(0,2*pi, length.out = n)
raio=1; x=raio*cos(theta); y=raio*sin(theta)
xt=c(x+1.*raio); yt=c(y)
n=16; rotacao=pi/n*1:(2*n); n=length(xt)
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]))
}
dt1=tibble::tibble(xt1,yt1)
theta=seq(0,2*pi, length.out = n); raio=2.25
x=raio*cos(theta); y=raio*sin(theta)
xt=c(x); yt=c(y)
dt=tibble::tibble(xt,yt)
p=ggplot()+coord_fixed()+ theme_void()
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color='red',size=.05)
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color='blue',size=.05)
p
```

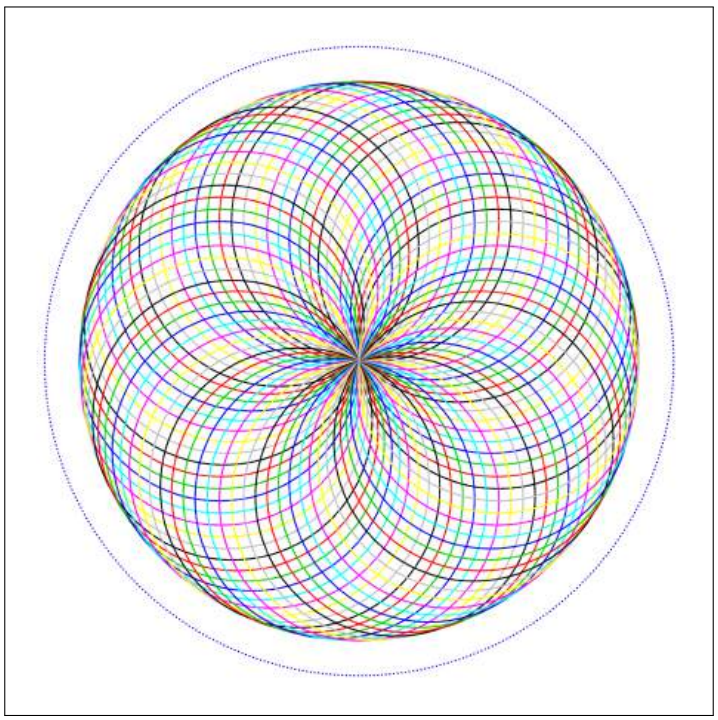
6.1 Variação de cores e efeitos

Aqui, uma pequena modificação nas estruturas de fluxo do tipo laço (*loop*) são realizadas para incorporar a cor e, por consequência, tornar possível modificação em cada elemento i do laço. A escolha das cores é arbitrária dentre aquelas cores disponíveis no R por meio do comando $colors()$.

6.1.1 Múltiplas cores

A Figura 6.5 mostra os resultados para $n = 36$ rotações com cores distintas. Note que apenas uma pequena alteração no código é executada: a inicialização da janela gráfica e a definição da cor, com valor i, para cada rotação i.

Figura 6.5: Composição de $n = 36$ rotações coloridas.



Fonte: Os autores.

Deve ser notado que há uma gama de possibilidades de explorações que podem ser realizadas, pois outras combinações de cores resultam em diferentes figuras. Como exemplo, $color = colors()[2 * i + 1]$ ou $color = colors()[i]$.

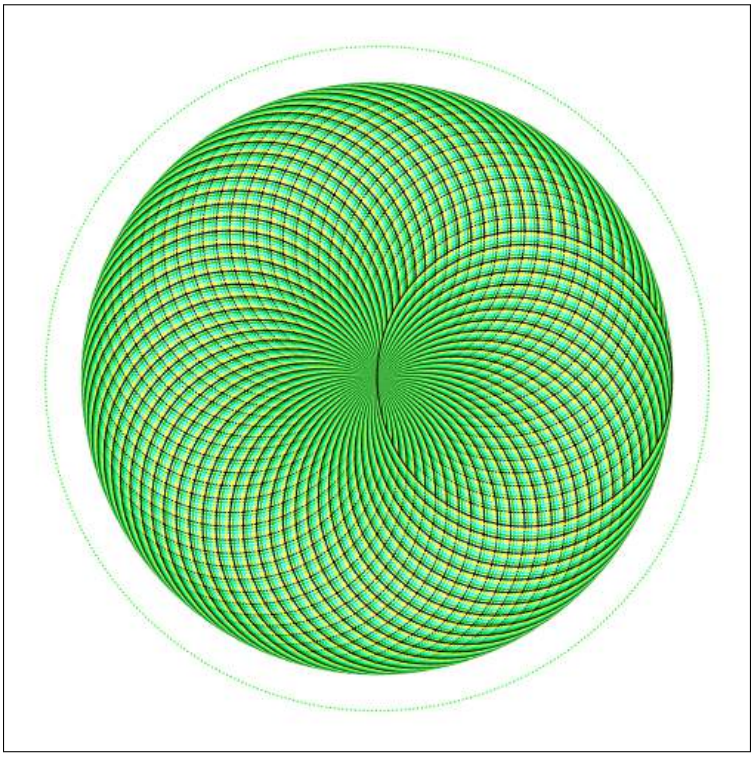
O código completo é apresentado a seguir. Os elementos descritivos são idênticos aos apresentados anteriormente.

A Figura 6.6 mostra uma modificação substancial no resultado para um número de rotações igual a $n = 136$. Neste caso, as cores foram escolhidas associadas ao parâmetro i que governa o *laço for* por meio de $color = 2 * i + 1$.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

```
require(ggplot2); n=500; theta=seq(0,2*pi, length.out = n)
raio=1; x=raio*cos(theta); y=raio*sin(theta); xt=c(x+1.*raio); yt=c(y)
n=36;  rotacao=pi/n*1:(2*n); n=length(xt); xt1=xt; yt1=yt
p=ggplot()+coord_fixed()+ theme_void()
colors=1:n
for(i in 1:length(rotacao)){
xt1=c(xt[1:n]*cos(rotacao[i])-yt[1:n]*sin(rotacao[i]))
yt1=c(xt[1:n]*sin(rotacao[i])+yt[1:n]*cos(rotacao[i]))
dt1=tibble::tibble(xt1,yt1)
p=p+ geom_point(data=dt1, aes(x=xt1, y=yt1), color=i,size=.05)}
theta=seq(0,2*pi, length.out = n); raio=2.25;
x=raio*cos(theta); y=raio*sin(theta); xt=c(x); yt=c(y)
dt=tibble::tibble(xt,yt)
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color='blue',size=.05)
p
```

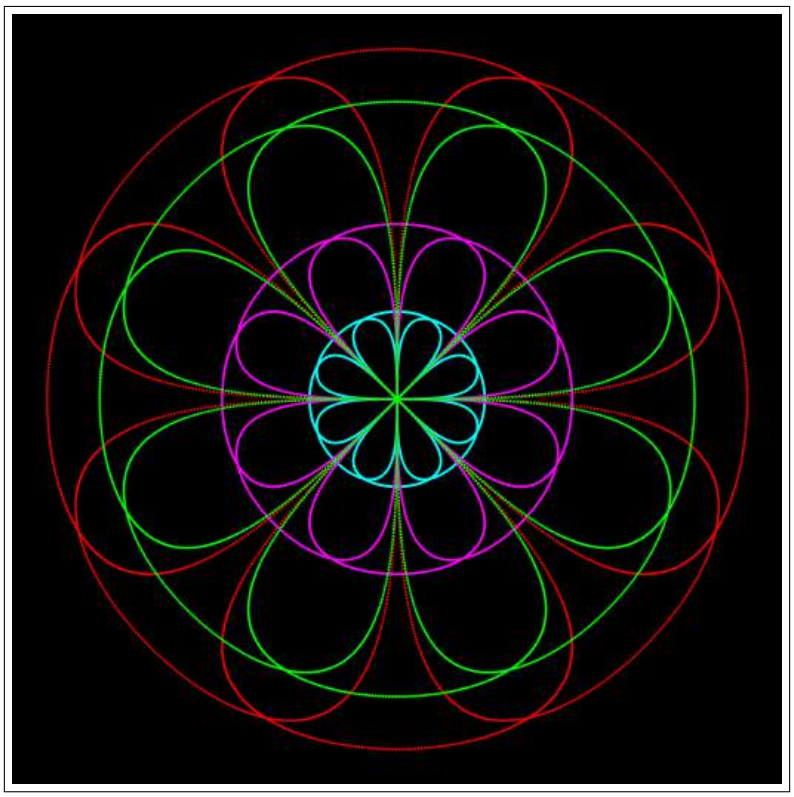
Figura 6.6: Composição de $n = 136$ rotações coloridas.



Fonte: Os autores.

Efeitos interessantes podem ser obtidos pela interação entre o plano de fundo e cores adotadas para ressaltar as características da curva paramétrica. A Figura 6.7 considera a rotação da lemniscata de Gerono com homotetias. As rotações formam um único conjunto de pontos que são coloridos com mesma cor, enquanto que as homotetias distintas são coloridas por diferentes cores.

Figura 6.7: Cor de fundo, rotações e homotetias em cores distintas.



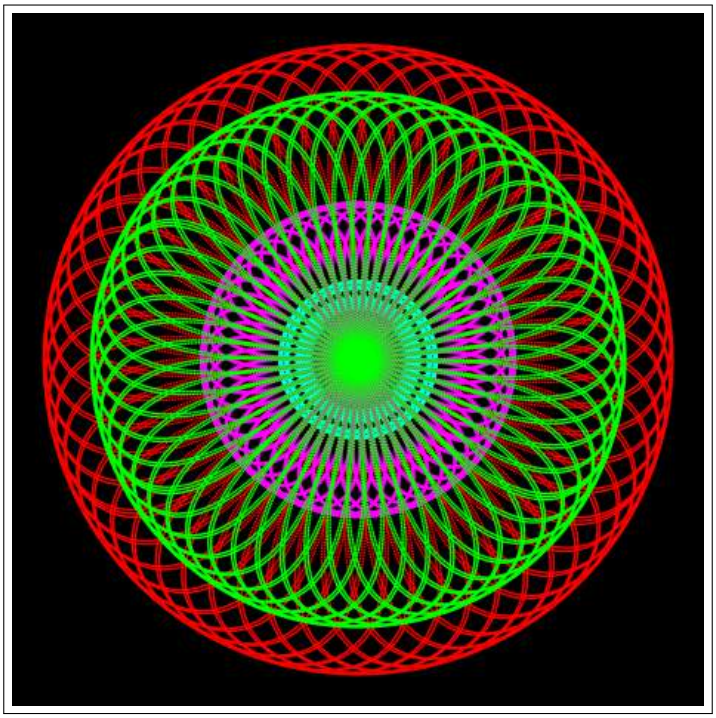
Fonte: Os autores.

O código completo é mostrado a seguir. Note que apenas as definições de plano de fundo e cores para cada homotetia é modificada, ou seja, as linhas de 18 a 25.

```
1   require(ggplot2); n=700; theta=seq(0,2*pi, length.out = n)
2   x=sin(theta); y=sin(theta)*cos(theta); z=rep(0,n)
3   rotacao=c(pi/4,pi/2,3*pi/4,pi, 3*pi/2); xt=x; yt=y
4   for(i in 1:length(rotacao)){
5   xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
6   yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i])) }
7   dt=tibble::tibble(xt,yt)
8   contracao = 0.25;xt2=xt*contracao; yt2=yt*contracao; dt2=tibble::tibble(xt2,yt2);
9   contracao = 0.5;xt3=xt*contracao;yt3=yt*contracao; dt3=tibble::tibble(xt3,yt3)
10  contracao = 0.85;xt4=xt*contracao;yt4=yt*contracao; dt4=tibble::tibble(xt4,yt4);
11  size=0.25
12  p= ggplot()+ coord_fixed()+ theme_void()
13  p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)
14  p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color="cyan",size=size)
15  p = p+geom_point(data=dt3, aes(x=xt3, y=yt3), color="magenta",size=size)
16  p = p+geom_point(data=dt4, aes(x=xt4, y=yt4), color="green",size=size)
17  p=p +theme(panel.background = element_rect(fill = "black")  )
18  p
```

Considere que a leminiscata de Gerono é rotacionada por uma quantidade arbitrária, *rotacao*, e os pontos agrupados nos vetores xt, yt. Esse caso é análogo ao caso anterior, com aumento significativo do número de pontos. A Figura 6.8 ilustra o resultado.

Figura 6.8: Plano de fundo, rotações e homotetias com cores distintas.



Fonte: Os autores.

O trecho de código a seguir ilustra a modificação efetuada no código anterior (linha 5)

```
1        require(ggplot2)
2        n=700;
3        theta=seq(0,2*pi, length.out = n)
4        x=sin(theta);
5        y=sin(theta)*cos(theta);
6        z=rep(0,n);
7        rotacao=c(seq(0,2*pi,0.125))
8        xt=x; yt=y
9        for(i in 1:length(rotacao)){
10       xt=c(xt,x[1:n]*cos(rotacao[i])-y[1:n]*sin(rotacao[i]))
11       yt=c(yt,x[1:n]*sin(rotacao[i])+y[1:n]*cos(rotacao[i]))
12       }
```

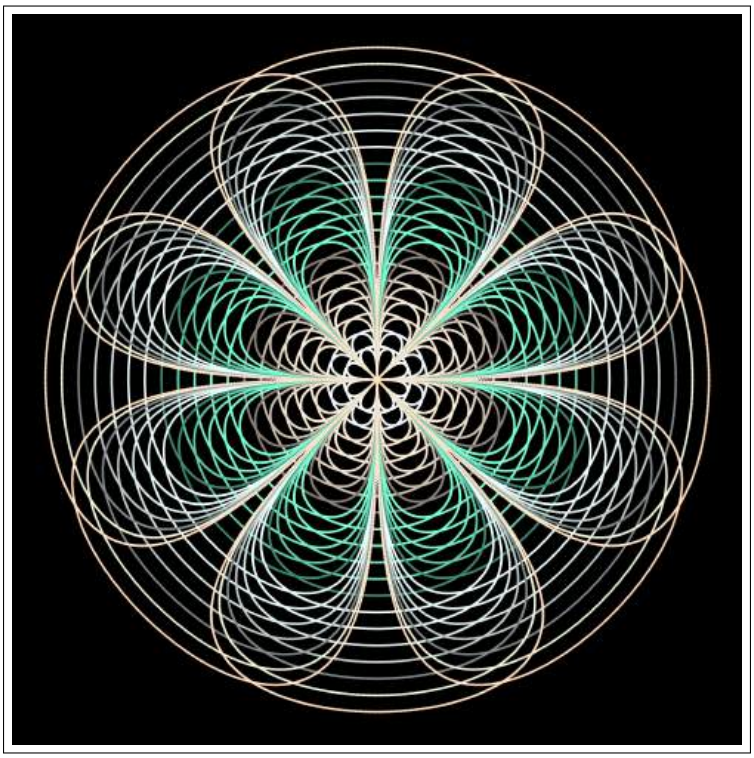
Deve ser notado as homotetias e respectivas cores são calculadas ou definidas manualmente. Este processo pode ser ineficiente para englobar um número arbitrário de operações de rotações ou homotetias.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

6.2 Número de homotetias e rotações arbitrários

Ao considerar números de rotações e homotetias arbitrários, é necessário modificar as estruturas de fluxo de forma a englobar essa variação. A Figura 6.9 mostra um exemplo em que o número de rotações é $nrot = 5$ e o número de homotetias, *contracao*, é definido por meio da sequência $contracao = seq(.1, 1, by = step)$ com $step = 0.05$.

Figura 6.9: Homotetias sucessivas da composição de rotações da lemniscata.



Fonte: Os autores.

As cores associadas por meio do índice $i = 1, \cdots, 19$ aplicados à função *colors*() são destacadas a seguir:

```
[1]   "white"          "aliceblue"      "antiquewhite"   "antiquewhite1"
[5]   "antiquewhite2"  "antiquewhite3"  "antiquewhite4"  "aquamarine"
[9]   "aquamarine1"    "aquamarine2"    "aquamarine3"    "aquamarine4"
[13]  "azure"          "azure1"         "azure2"         "azure3"
[17]  "azure4"         "beige"          "bisque"
```

As cores anteriores são utilizadas na ordem em que aparecem, por exemplo, a cor `"white"` é utilizada para colorir a primeira homotetia de razão *contracao*[1] e assim sucessivamente.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

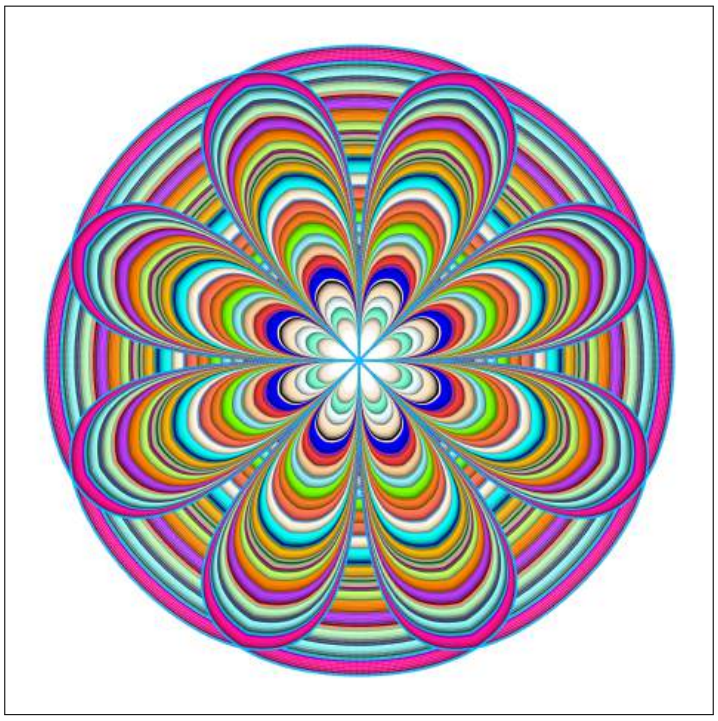
6.3 Efeito de preenchimento

O efeito de preenchimento pode ser obtido ao redefinir o número de homotetias. A Figura 6.10 mostra um exemplo com 5 rotações dadas por $rotacao = c(pi/4, pi/2, 3*pi/4, pi, 3*pi/2)$ e homotetias dadas por $contracao = seq(.1, 1, by = step)$ com com $step = 0.0075$. Neste caso, o número de homotetias é 121 e, portanto, as primeiras 121 cores distintas de $colors()$ foram utilizadas na coloração da figura. A seguir estão as 20 últimas cores utilizadas no processo.

```
 [1] "darkseagreen"   "darkseagreen1"  "darkseagreen2"  "darkseagreen3"
 [5] "darkseagreen4"  "darkslateblue"  "darkslategray"  "darkslategray1"
 [9] "darkslategray2" "darkslategray3" "darkslategray4" "darkslategrey"
[13] "darkturquoise"  "darkviolet"     "deeppink"       "deeppink1"
[17] "deeppink2"      "deeppink3"      "deeppink4"      "deepskyblue"
```

As Figuras 6.11, 6.12, 6.13 e 6.14 mostram variações obtidas apenas pela variação do número de rotações. A diferença entre os códigos são apenas na definição do número de rotações e substituição da definição manual para uma definição em termos de sequência no intervalo $[0, \pi]$. Em todos os casos mostrados, a sobreposição das diferentes cores ocasiona um efeito de entrelaçamento interessante.

Figura 6.10: Efeito de preenchimento em rotações sem interseções.



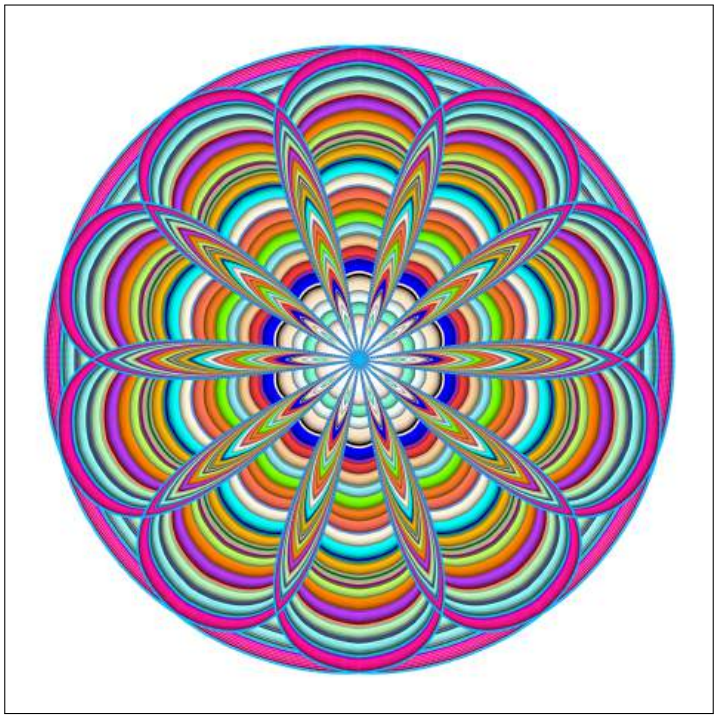
Fonte: Os autores.

O código completo é mostrado a seguir.

```
require(ggplot2);
n=1000;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)
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)
step=0.0075; contracao = seq(.1,1,by=step);size=0.25
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) }
p
```

A Figura 6.11 utiliza $nrot = 5$ no intervalo $[0, \pi]$. Note que a sobreposição das rotações gera um novo padrão de sobreposição.

Figura 6.11: Composição com $nrot = 5$ rotações no intervalo $[0, \pi]$.

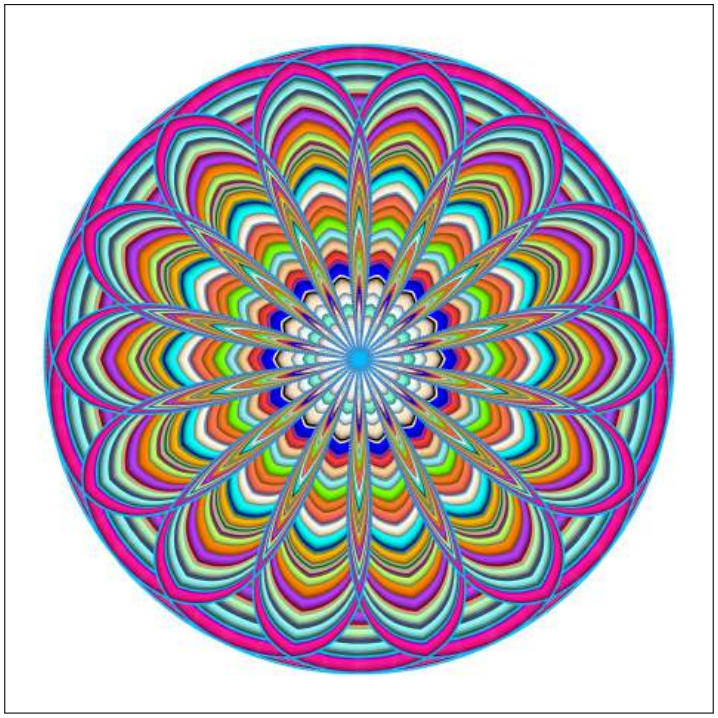


Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

A Figura 6.12 utiliza $nrot = 7$ no intervalo $[0, \pi]$.

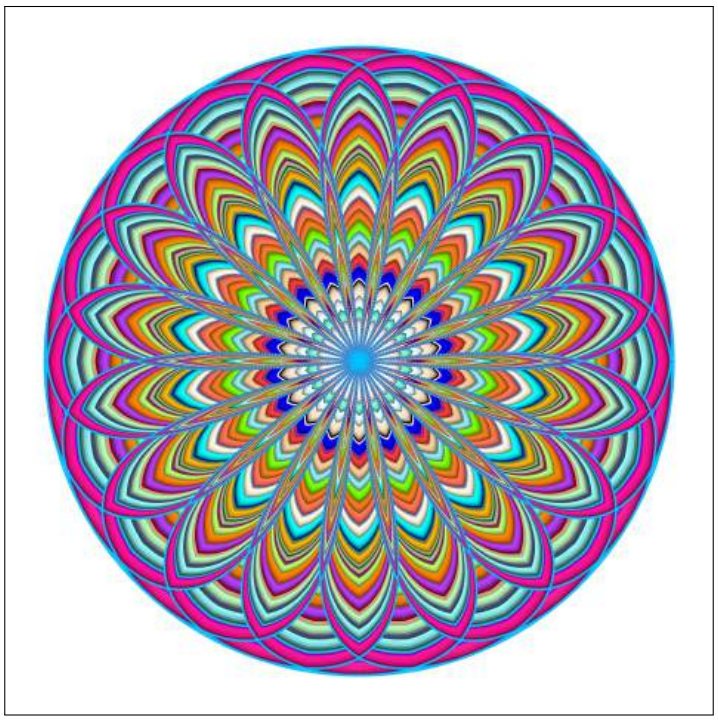
Figura 6.12: Composição com $nrot = 7$ rotações no intervalo $[0, \pi]$.



Fonte: Os autores.

A Figura 6.13 utiliza $nrot = 9$ no intervalo $[0, \pi]$.

Figura 6.13: Composição com $nrot = 9$ rotações no intervalo $[0, \pi]$.

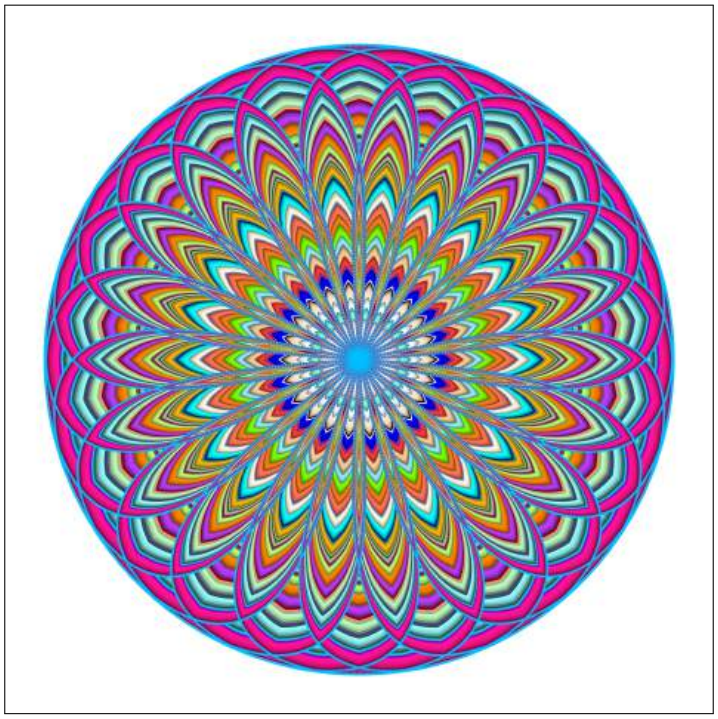


Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

A Figura 6.14 utiliza $nrot = 11$ no intervalo $[0, \pi]$.

Figura 6.14: Composição com $nrot = 11$ rotações no intervalo $[0, \pi]$.



Fonte: Os autores.

As explorações anteriores permitem deduzir a existência de um vasto número de possibilidades de combinações de cores e efeitos, pois é possível realizar a combinação de diferentes números de rotações e diferentes passos para a discretização dos intervalos que definem as homotetias.

As Figuras 6.15 até 6.23 ilustram alguns resultados interessantes, obtidos por diferentes combinações de cores, número de rotações e número de homotetias. Analogamente aos casos anteriores, o número de cores utilizado é igual ao número de homotetias e, portanto, o padrão observado é diretamente relacionado ao passo, *step*, adotado. Modificações no passo afetam diretamente o número de cores que será utilizado; modificações na definição dos valores inicial e final da sequência de homotetias, *contracao*, também ocasiona mudanças no padrão observado.

A Figura 6.15 utiliza $nrot = 13$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$ e homotetias $contracao = seq(.1, 1., by = step)$ com $step = 0.005$.

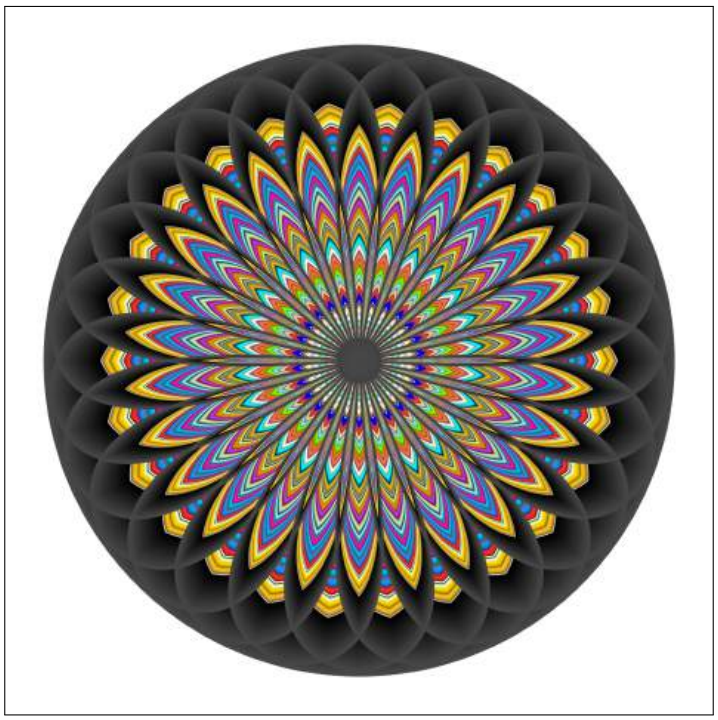
A Figura 6.16 utiliza $nrot = 3$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$ e homotetias $contracao = seq(.1, 1., by = step)$ com $step = 0.0025$.

A Figura 6.17 utiliza $nrot = 27$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.1, 1.5, by = step)$ com $step = 0.0125$.

A Figura 6.18 utiliza $nrot = 13$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.1, 1., by = step)$ com $step = 0.0125$.

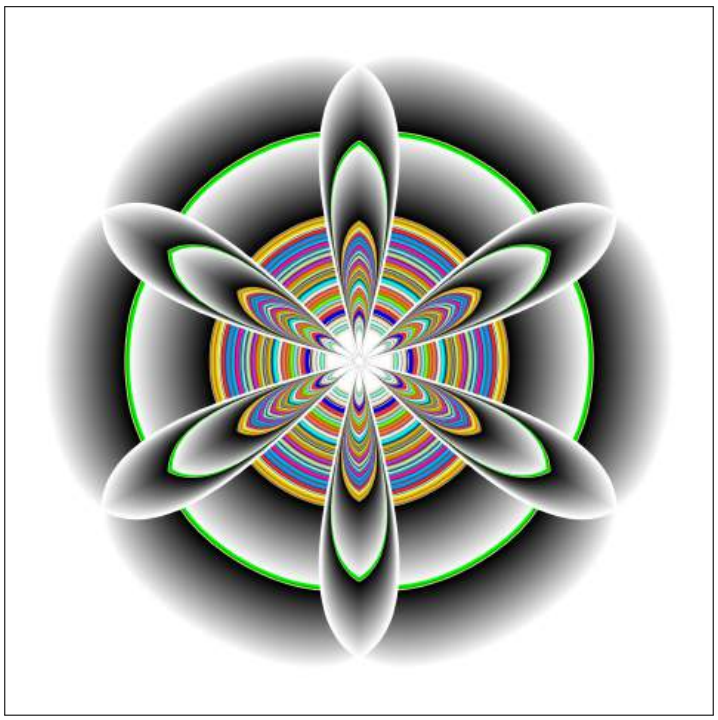
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.15: Variações de cinza, "gray-"gray28", predominante nos extremos.



Fonte: Os autores.

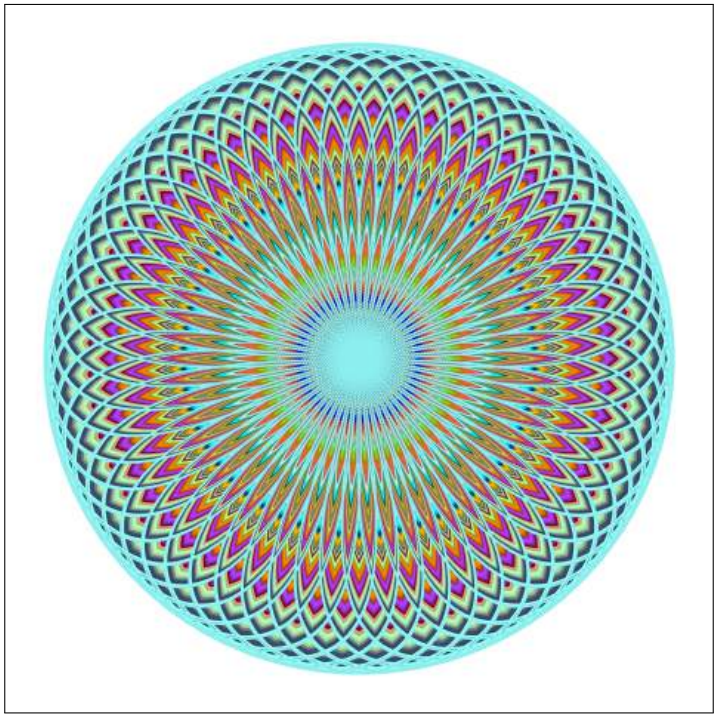
Figura 6.16: Variações de cinza, até "gray100", predominante nos extremos.



Fonte: Os autores.

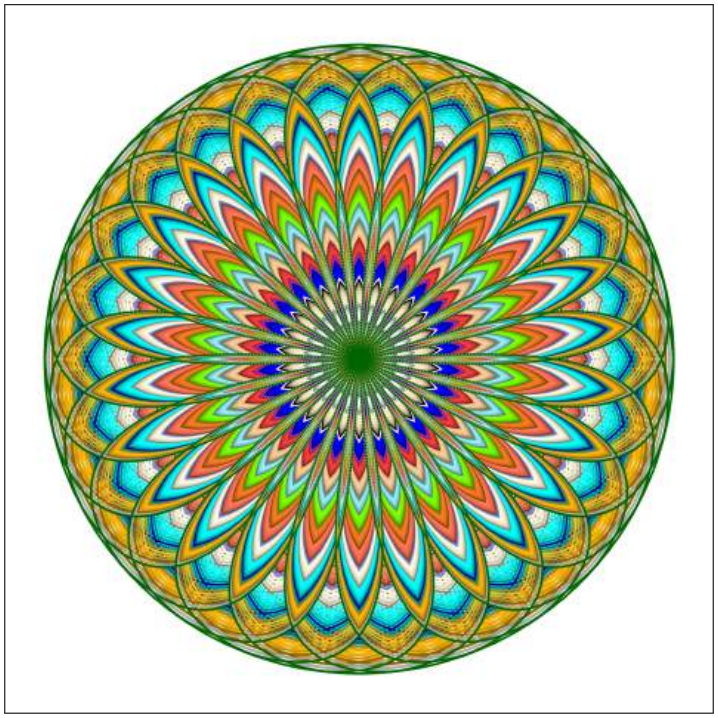
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.17: Composição com cores entre "white"e "darkslategray2".



Fonte: Os autores.

Figura 6.18: Composição com cores entre "white"e "darkgreen".

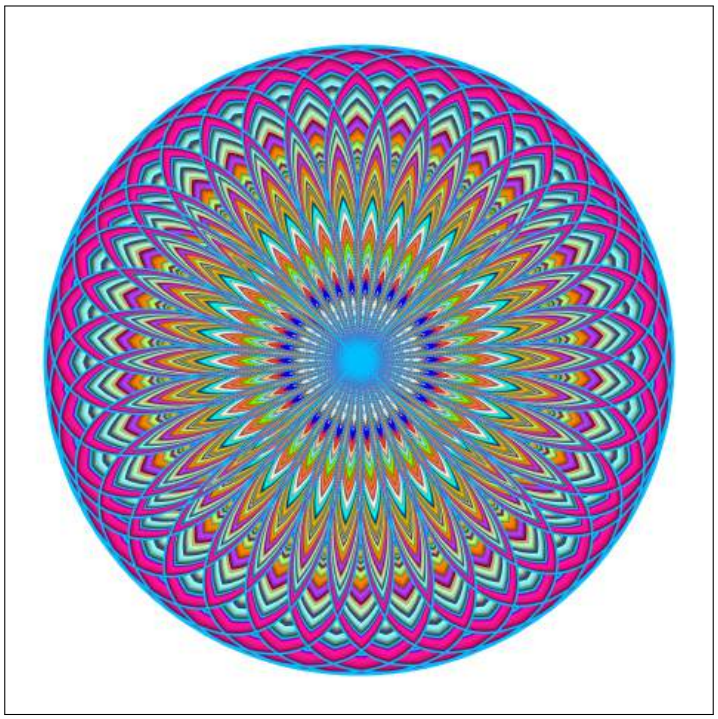


Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

A Figura 6.19 utiliza $nrot = 17$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.1, 1., by = step)$ com $step = 0.0075$.

Figura 6.19: Composição com cores entre "white"e "deepskyblue".



Fonte: Os autores.

As Figuras a seguir mostram variações dos extremos utilizados para o cálculo da sequência de homotetias.

A Figura 6.20 utiliza $nrot = 13$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.1, .5, by = step)$ com $step = 0.00175$. Neste caso, $n = 286$ cores variando entre "white"e "grey25"são utilizadas no processo de construção.

A Figura 6.21 utiliza $nrot = 13$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.1, .6, by = step)$ com $step = 0.00175$. Neste caso, $n = 343$ cores variando entre "white"e "grey82"são utilizadas no processo de construção.

A Figura 6.22 utiliza $nrot = 23$ em$[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.3, .7, by = step)$ com $step = 0.00175$. Neste caso, $n = 286$ cores variando entre "white"e "grey25"são utilizadas no processo de construção.

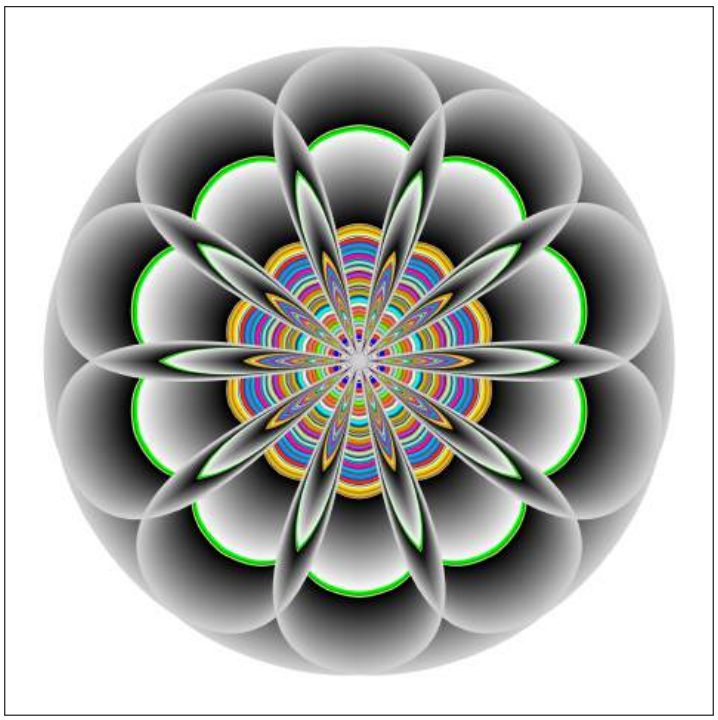
A Figura 6.23 utiliza $nrot = 23$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.2, .7, by = step)$ com $step = 0.00175$. Neste caso, $n = 343$ cores variando entre "white"e "grey82"são utilizadas no processo de construção.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.20: Composição com cores entre "white"e "grey25".

Fonte: Os autores.

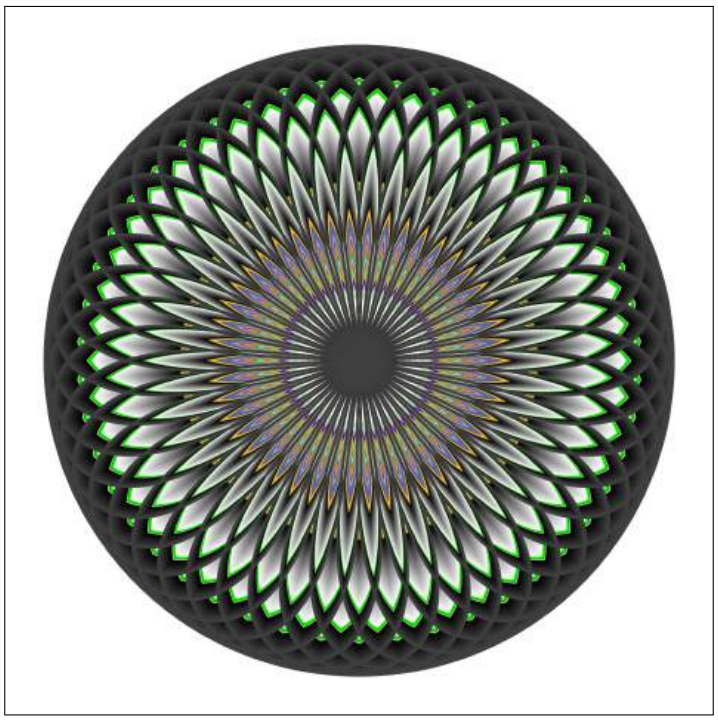
Figura 6.21: Composição com cores entre "white"e "grey82".



Fonte: Os autores.

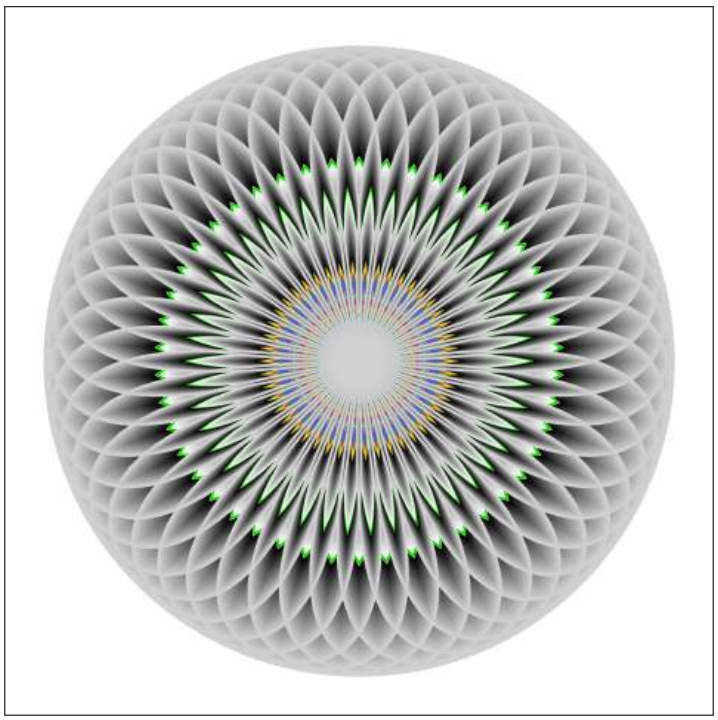
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.22: Composição com cores entre "white"e "grey25".



Fonte: Os autores.

Figura 6.23: Composição com cores entre "white"e "grey82".



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Até o momento, as cores são definidas de forma sequencial com base na disponibilidade da função $colors()$. A seguir, escolhas particulares das cores disponíveis em $colors()$ são utilizadas. Ainda, o número de cores escolhido deve ser idêntico ao comprimento do vetor de iteração das homotetias ou um ajuste deve ser feito para que haja igualdade. Por exemplo, suponha que as cores $colors()[1:2]$, "white"e "aliceblue", sejam escolhidas, mas o vetor de homotetias possua comprimento 5, $contracao = c(.1, .2, .3, .4, .5)$, então é necessário reajustar o vetor de cores para que possua tamanho cinco; uma possível solução é replicar o vetor de cores escolhido; outra solução é amostrar o número de cores desejado do conjunto de possibilidades. Aqui, o processo de amostragem com reposição é adotado por conveniência.

As Figuras 6.24 até 6.30 ilustram escolhas específicas de cores com $nrot = 15$ rotações e $set.seed(10)$. A adaptação do código, mostrado ao fim da seção, é apenas na parte relativa à escolha das cores a serem utilizadas.

A Figura 6.24 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas de $colors()[1:27]$ com reposição.

A Figura 6.25 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas de $colors()[300:361]$ com reposição.

A Figura 6.26 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas $colors()$ com reposição.

A Figura 6.27 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas de $colors()[142:361]$ com reposição.

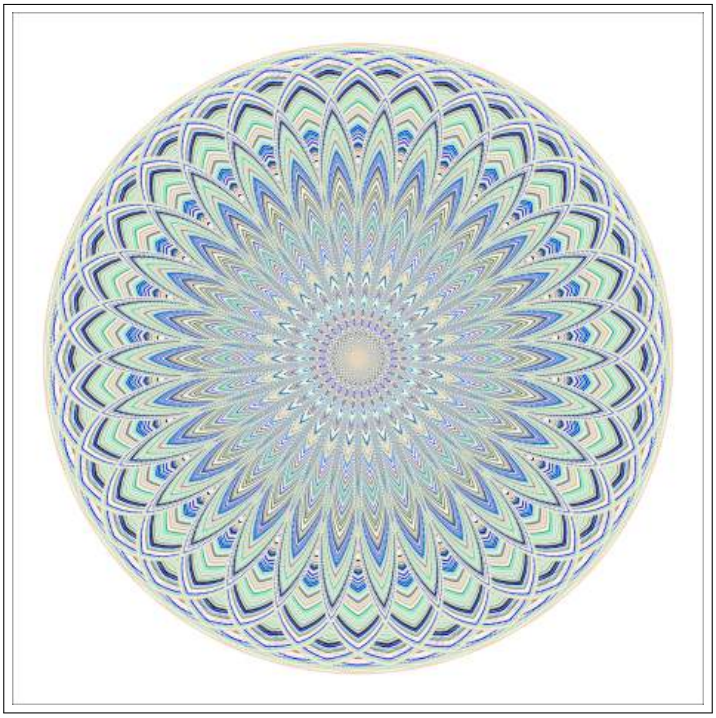
A Figura 6.28 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas de $colors()[142:156]$.

A Figura 6.29 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas de $colors()[142:151]$ com reposição.

A Figura 6.30 utiliza $nrot = 15$ em $[0, \pi]$, rotações $rotacao = seq(0, pi, pi/nrot)$, homotetias $contracao = seq(.0, 1.75, by = step)$ com $size = 0.015$ e cores extraídas de por $colors()[200:300]$ com reposição.

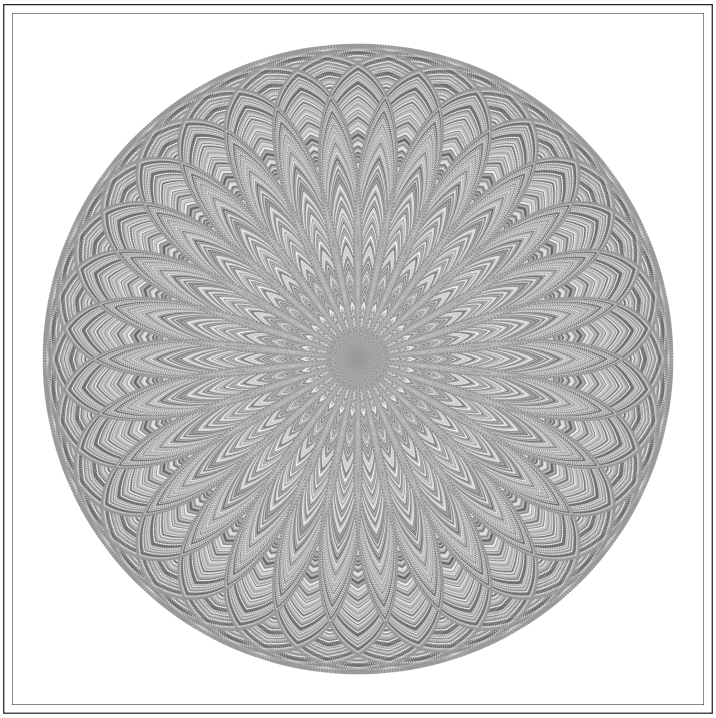
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.24: Composição com cores $colors()[1:27]$.



Fonte: Os autores.

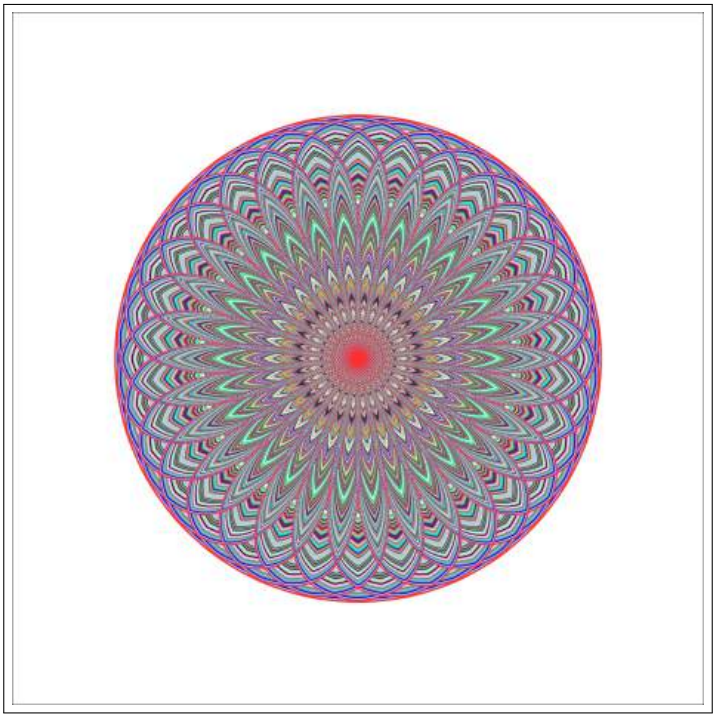
Figura 6.25: Composição com cores $colors()[300:361]$.



Fonte: Os autores.

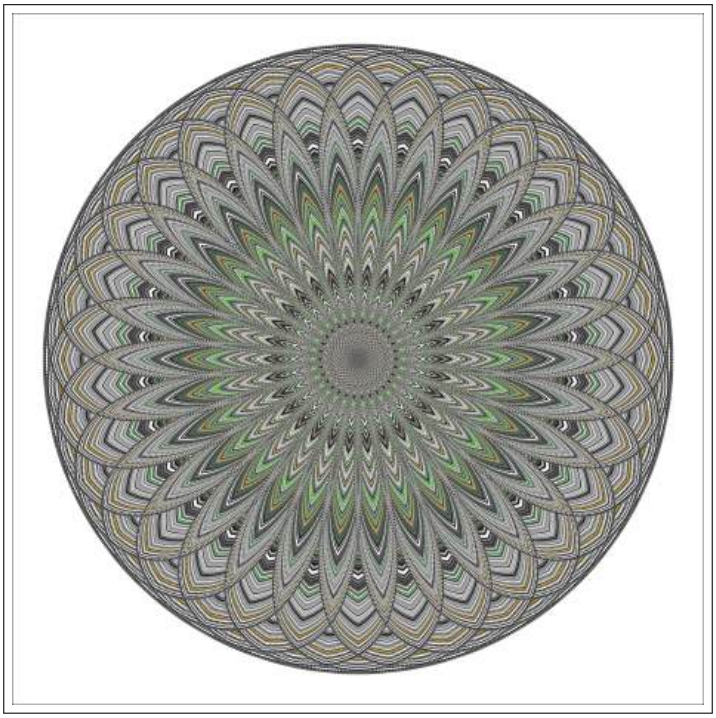
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.26: Composição com cores $colors()$.



Fonte: Os autores.

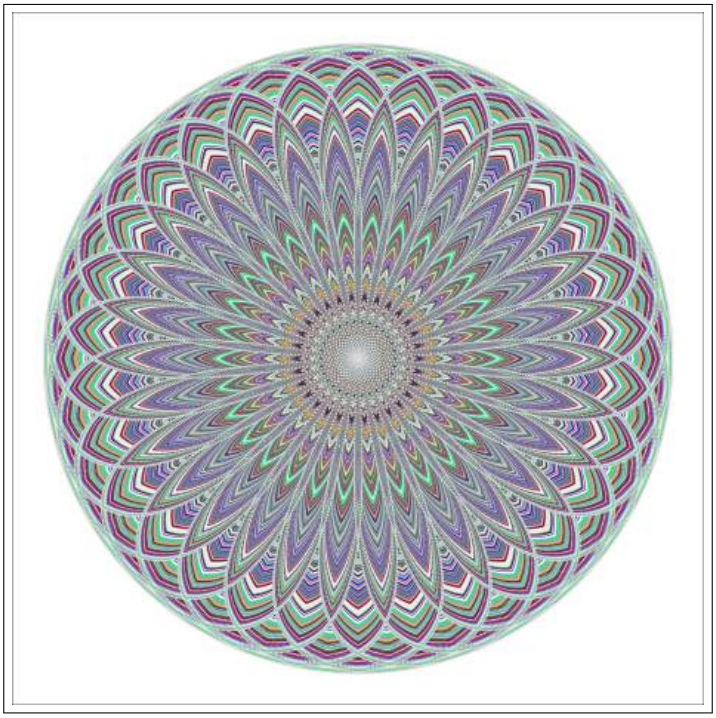
Figura 6.27: Composição com cores $colors()[142:361]$.



Fonte: Os autores.

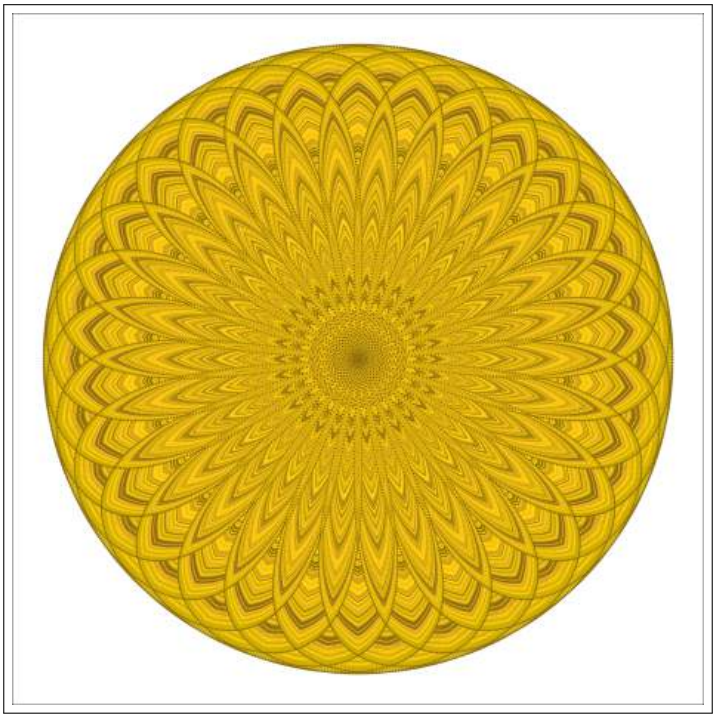
ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.28: Composição com cores $colors()[142:156]$.



Fonte: Os autores.

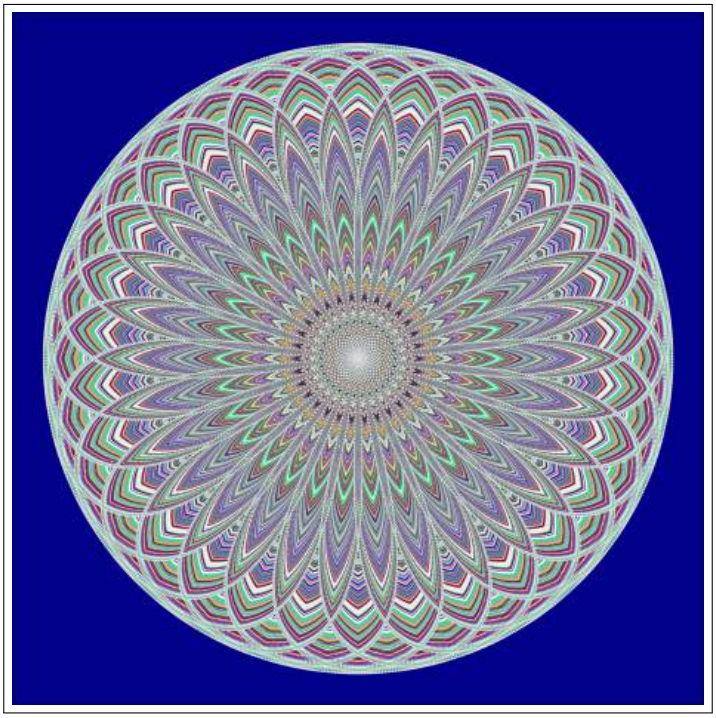
Figura 6.29: Composição com cores $colors()[142:151]$.



Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Figura 6.30: Composição com $colors()[200:300]$.



Fonte: Os autores.

```
require(ggplot2); n=700; theta=seq(0,2*pi, length.out = n)
x=sin(theta); y=sin(theta)*cos(theta); z=rep(0,n)
nrot=15
rotacao=c(seq(0.0,pi,pi/nrot))#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); size=.025
p=p+ geom_point(data=dt, aes(x=xt, y=yt), color="red",size=size)
step=0.0075
contracao = seq(.0,1.75,by=step);size=0.015
set.seed(10)
#cores=sample(colors()[1:27],length(contracao),replace=T)   #Figura 6.24
#cores=sample(colors()[300:361],length(contracao),replace=T)#Figura 6.25
#cores=sample(colors(),length(contracao),replace=T)         #Figura 6.26
#cores=sample(colors()[142:361],length(contracao))          #Figura 6.27
#cores=sample(colors()[142:156],length(contracao),replace=T)#Figura 6.28
#cores=sample(colors()[142:151],length(contracao),replace=T)#Figura 6.29
cores=sample(colors()[200:300],length(contracao),replace=T)#Figura 6.30
step=0.0175
MinhasCores=rep(colores,floor(length(contracao)/length(colores)))
```

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

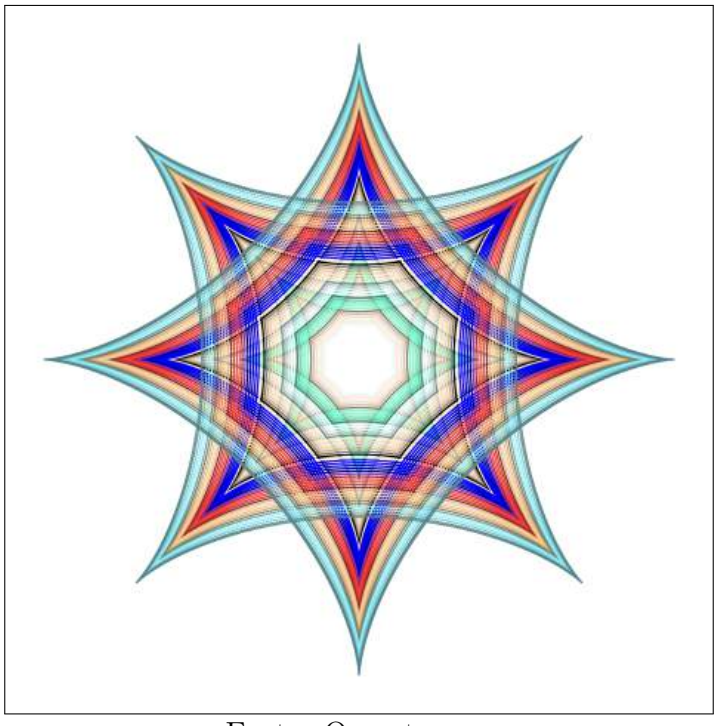
```
24        for(i in 1:length(contracao)){
25        xt2=c(xt*contracao[i])
26        yt2=c(yt*contracao[i])
27        dt2=tibble::tibble(xt2,yt2)
28        p=p+geom_point(data=dt2, aes(x=xt2, y=yt2),color=MinhasCores[i],size=size)
29        p }
30        p=p +theme(panel.background = element_rect(fill = "darkblue")  )
31        p
```

6.4 Astroide, deltoide e lemniscata de Bernoulli

O processo apresentado anteriormente pode ser aplicado em outras curvas planas. No entanto, não há garantias que todas as curvas produzam resultados interessantes. Em alguns casos, ajustes serão necessários; em outros mudanças na forma de coloração e alterações nas posições podem ser necessários. A seguir, estão figuras geradas com o astroide, o deltoide e a lemniscata de Bernoulli.

A Figura 6.31, selecionadas dentre aquelas disponíveis em *colors*(), mostra a composição com apenas uma rotação. Note que não há restrição do número de cores e nem do número de rotações. Mais rotações criam uma figura com aspecto circular central em torno da origem. Da mesma forma que na seção anterior, cores específicas podem ser utilizadas na composição.

Figura 6.31: Composição do astroide com cores determinadas pelas homotetias.

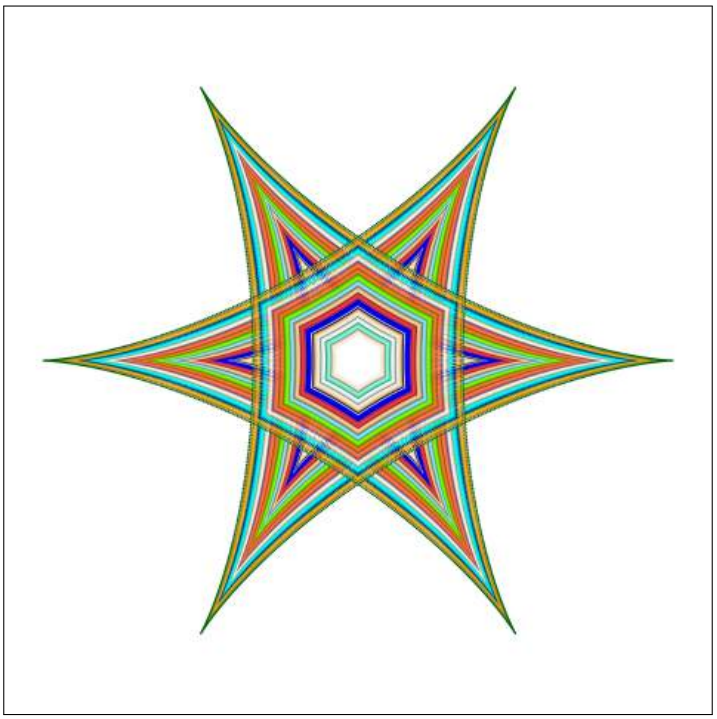


Fonte: Os autores.

```
require(ggplot2)
n=1000; theta=seq(0,2*pi, length.out = n)
x=cos(theta)^3; y=sin(theta)^3
z=rep(0,n); dt=tibble::tibble(x,y,z)
step=pi/4;rotacao=c(seq(0,pi,step)); xt=x; yt=y
p=ggplot()+coord_fixed()+theme_void()
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]))}
red=c(seq(0.2,1,0.0175))#c(.125,0.25, 0.5, 0.75,.9,.95,.975,1.)
for(i in 1:length(red)){
xtt=c(xt*red[i]); ytt=c(yt*red[i])
dt=data.frame(x=c(xt, xtt), y=c(yt, ytt), z="astroide")
p=p+geom_point(data=dt, aes(x=x, y=y), color=colors()[i],size=0.05) }
p
```

A Figura 6.32 é um outro exemplo de composição e, novamente, não há restrição do número de cores e nem do número de rotações. A figura de aspecto circular central ainda será obtida para um número de rotações elevado.

Figura 6.32: Composição do deltoide com cores determinadas pelas homotetias.



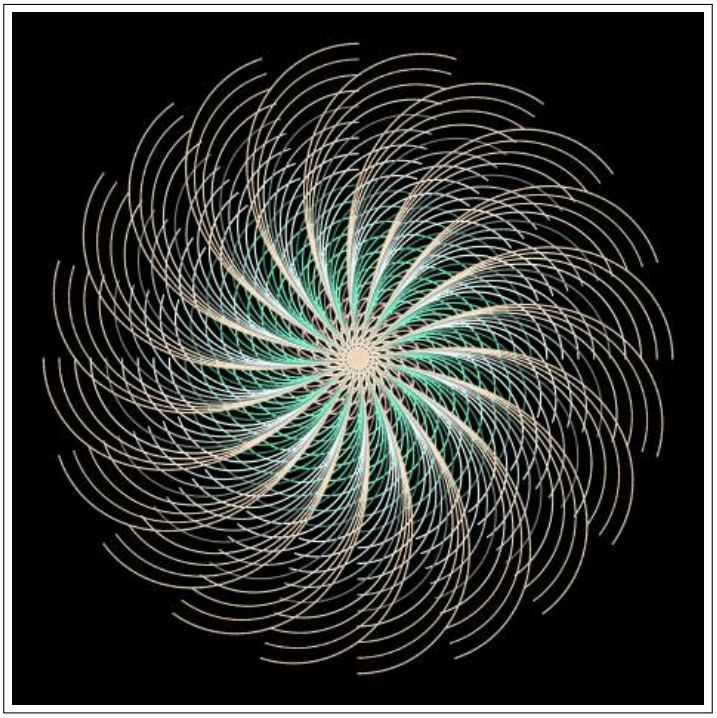
Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

```
require(ggplo2)
n=1000; theta=seq(0,2*pi, length.out = n)
x=2*cos(theta)+cos(2*theta)
y=2*sin(theta)-sin(2*theta)
z=rep(0,n); dt=tibble::tibble(x,y,z)
step=pi; rotacao=c(seq(0,pi,step)); xt=x; yt=y
p=ggplot()+coord_fixed()+theme_void()
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]))}
red=c(seq(0.2,1,0.01))
for(i in 1:length(red)){
    xtt=c(xt*red[i]); ytt=c(yt*red[i])
    dt=data.frame(x=c(xt, xtt), y=c(yt, ytt), z="astroide")
    p=p+geom_point(data=dt, aes(x=x, y=y), color=colors()[i],size=0.05) }
p
```

A Figura 6.33 é um outro exemplo de composição, mas agora a lemniscata de Bernoulli. É resultado de um certo número de rotações e homotetias. As cores utilizadas foram aquelas disponíveis em *colors*, escolhidas a partir da primeira cor, e determinadas pelo comprimento do vetor de homotetias.

Figura 6.33: Composição da lemniscata de Bernoulli parcial com cores determinadas pelas homotetias.



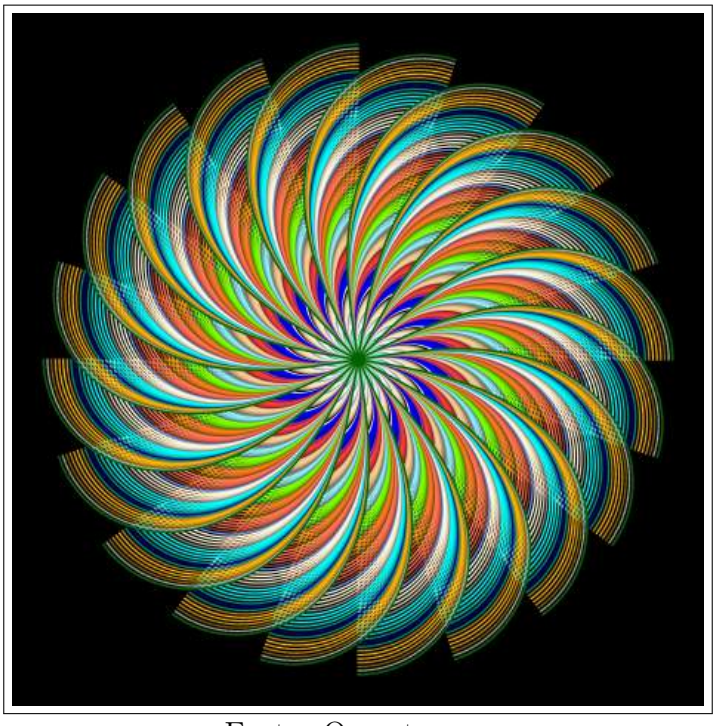
Fonte: Os autores.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

```
require(ggplo2)
n=500; theta=seq(0,pi, length.out = n)
x=cos(theta)/(1+(sin(theta))^2);
y=sin(theta)*cos(theta)/(1+(sin(theta))^2)
z=rep(0,n); dt=tibble::tibble(x,y,z)
step=pi/10
rotacao=c(seq(0,pi,step));
xt=x; yt=y
p=ggplot()+coord_fixed()+theme_void()
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]))}
        red=c(seq(0.,1,0.05))#c(.125,0.25, 0.5, 0.75,.9,.95,.975,1.)
        for(i in 1:length(red)){
        xtt=c(xt*red[i])
        ytt=c(yt*red[i])
        dt=data.frame(x=c(xt, xtt), y=c(yt, ytt), z="astroide")
        p=p+geom_point(data=dt, aes(x=x, y=y), color=colors()[i],size=0.05)}
        p=p + theme(panel.background = element_rect(fill = "black")  )
        p
```

Finalmente, a Figura 6.34 é um exemplo de resultado gerado pelo aumento do número de homotetias.

Figura 6.34: Homotetias sucessivas da composição de rotações:



Fonte: Os autores.

Capítulo 7

Considerações Finais

Apesar da linguagem R ter sido criada para resolver problemas de análise de dados por meio de métodos estatísticos, sua utilização foi explorada em vários outros contextos devido à existência de uma grande comunidade de usuários em diferentes áreas do conhecimento. Este livro é um exemplo de aplicação do R em uma situação "pouco comum", por se tratar de um software estatístico. Ilustra, além da flexibilidade dessa linguagem de programação, a evolução dos pacotes gráficos desenvolvidos colaborativamente pela comunidade de usuários.

Do desafio de construir mandalas usando R, surgiram vários questionamentos sobre o processo de construção como, por exemplo, a escolhas das curvas, como gerar essas curvas, o algoritmo para a construção e, finalmente, a estética do produto final, a mandala. Os códigos apresentados no texto são frutos de algumas escolhas, dentre várias outras possibilidades testadas. A sequência de desenvolvimento surgiu da concepção de realizar a apresentação passo a passo, iniciando com o traçado das curvas e evoluindo até a composição das figuras, explicitando as etapas intermediárias. O processo de coloração foi um pouco mais elaborado e ocorreu em uma sequência crescente de complexidade para atingir os resultados dos padrões que foram obtidos na construção de cada Mandala.

As construções apresentadas no texto não excluem as possibilidades de utilização de quadrados, retângulos, polígonos regulares ou qualquer outra curva de interesse. Nestes casos, a aplicação das transformações de rotação e homotetias é similar e envolve mínimos ajustes nos códigos. Também é necessário elaborar um pouco mais o processo matemático em alguns casos de transformações, tais como, as transformações afins ou rotações em torno de pontos diferentes da origem. Por exemplo, a rotação de um quadrado exige a escolha de um ponto para, em seguida, utilizar os elementos de rotação, translação, homotetias e respectivas colorações.

Ainda há espaço para explorar outras questões geométricas de translação e reflexão, por exemplo. Também há espaço para explorar conceitos combinatórios relacionados à coloração das figuras por meio de cores distintas ou não, quantidade de segmentos de retas que podem ser obtidos, etc. Outra possibilidade está na exploração de outras questões de otimização

de algoritmos ou de programação tais como aqueles apresentados na construção do cubo de Metatron. É possível ainda explorar conceitos de progressões aritméticas e geométricas que estão presentes nas construções das mandalas, além de outras funções geradoras ou da utilização de pacotes de cores pré definidos.

Finalizando, o material destaca-se pela produção de figuras de complexidade estrutural, as quais são provenientes da sobreposição de transformações geométricas de rotação e homotetias. Mas, ressalta-se que o número de curvas padrão ou figuras geométricas exploradas nesta obra, somadas ao número de possibilidades de padrões das cores aqui geradas é apenas pequena amostra das possibilidades existentes para a geração de Mandalas. Dessa forma, é provável que resultados adicionais possam ser incorporados futuramente. Com este apelo, considere estabelecido um desafio para que a criatividade ora despertada possa resultar em inesperados padrões geométricos a serem revelados no ambiente computacional do R.

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

Capítulo 8

Referências Bibliográficas

[1] ALCOFORADO, L. **Utilizando A Linguagem R: Conceitos, manipulação, visualização, modelagem e elaboração de relatórios.** Alta Books, 2021. ISBN 9786555201277. Disponível em: https://books.google.com.br/books?id=Um8ZEAAAQBAJ. (Seção 1, 2, 6)

[2] ALCOFORADO, L.; LEVY, A. **Visualização de Dados com o software R**. LFA, 2017. ISBN 9788592293208. Disponível em: http://www.estatisticacomr.uff.br/?page_id=21. (Seção 2)

[3] BEZERRA, J. **Mandalas**. Toda Materia, 2022. Disponível em: https://www.todamateria.com.br/mandala/. (Seção 1.2)

[4] FARIA, R.; MALTEMPI, M. Padrões fractais: conectando matemática e arte. **EccoS – Rev. Cient.**, v. 27, p. 33–53, 2012. (Seção 1.1)

[5] FERRéOL, R.; BOUREAU, S.; ESCULIER, A. 2d curves. 2017. Disponível em: https://mathcurve.com/courbes2d.gb/courbes2d.shtml. Acessado 29 abr.2022. (Seção 3.7)

[6] FRAZIER, M. **R Color CheatSheets**. National Center for Ecological Analysis and Synthesis-NCEAS, 2020. Disponível em: https://www.nceas.ucsb.edu/sites/default/files/2020-04/colorPaletteCheatsheet.pdf. (Seção 6)

[7] IRWIN, R. **A/r/tography: Rendering Self Through Arts-Based Living Inquiry**. [S.l.]: Pacific Educational Press, 2004. ISBN 9781895766707. (Seção 1.1)

[8] KEITT, T. **colorRamps: Builds Color Tables**. [S.l.], 2022. R package version 2.3.1. (Seção 6)

[9] LI, Y. et al. Research and trends in stem education: a systematic review of journal publications. **International Journal os STEM Educations**, v. 7, 2020. (Seção 1.1)

[10] LINARES, J. L.; SANTOS, J.; FIRMIANO, A. **Geometria Olímpica com GeoGebra**. [S.l.]: Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2022. v. 2. 115 p.ISBN 978-65-87023-23-6 (e-book). (Seção 1)

[11] MARTINS, M. M.; PICOSQUE, G.; GUERRA, M. T. Teoria e prática do ensino de arte: a língua do mundo. In: **Teoria e prática**. São Paulo: FTD: [s.n.], 2010. p. 208. (Seção 1.1)

[12] MÜLLER, K.; WICKHAM, H. **tibble: Simple Data Frames**. [S.l.], 2022. R package version 3.1.8. Disponível em: https://CRAN.R-project.org/package=tibble. (Seção 2.1)

[13] NEITZEL, A. d. A.; STEIL, I.; FRANCEZ, L. Contribuições da perspectiva metodológica "investigação baseada nas artes"e da a/r/tografia para as pesquisas em educação. **Educação em Revista**, v. 52, p. 158–178, 2022. Disponível em: https://seer.fundarte.rs.gov.br/index.php/RevistadaFundarte/issue/view/88/134. (Seção 1.1)

[14] OLIVEIRA, M. O.; CHARREU, L. Pesquisa educacional baseada em arte: A/r/tografia. **Revista da FUNDARTE**, v. 32, 2016. (Seção 1.1)

[15] O'CONNOR, J.; ROBERTSON, E. Famous Curves Index. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/. Acesso 26 out. 2022. (Seção 1)

[16] O'CONNOR, J.; ROBERTSON, E. Curves: Circle. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/Circle/. Acesso 10 abr. 2022. (Seção 3.2)

[17] O'CONNOR, J.; ROBERTSON, E. A history of pi. **MacTutor**, 2001. Disponível em: https://mathshistory.st-andrews.ac.uk/HistTopics/Pi_through_the_ages/. Acesso 10 abr. 2022. (Seção 3.2)

[18] O'CONNOR, J.; ROBERTSON, E. Curves: Elipse. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/Ellipse/. Acesso 18 abr. 2022. (Seção 3.3)

[19] O'CONNOR, J.; ROBERTSON, E. Curves: Cardioid. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/Cardioid/. Acesso 18 abr. 2022. (Seção 3.4)

[20] O'CONNOR, J.; ROBERTSON, E. Curves: Limacon of pascal. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/Limacon/. Acesso 18 abr. 2022. (Seção 3.9)

[21] O'CONNOR, J.; ROBERTSON, E. Curves: Astroid. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/Astroid/. Acesso 29 abr. 2022. (Seção 3.6)

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

[22] O'CONNOR, J.; ROBERTSON, E. Famous curves index. **MacTutor**, 2022. Disponível em: https://mathshistory.st-andrews.ac.uk/Curves/. Acesso 26 out. 2022. (Seção 1)

[23] PUCCETTI, R.; SOUZA, M. I. P. O. Diversidade e autoimagem: da representação à apresentação. In: . Londrina: Laboratório de Estudos dos Domínios da Imagem (LEDI), Universidade Estadual de Londrina-PR, 2011. p. 2507–2521. Disponível em: http://www.uel.br/eventos/eneimagem/anais2011/trabalhos/autores_M.htm. (Seção 1.1)

[24] R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2020. Disponível em: https://www.R-project.org/. (Seção 1)

[25] R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2023. Disponível em: https://www.R-project.org/. (Seção 6)

[26] STAUFFER, R. et al. Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations. **Bulletin of the American Meteorological Society**, v. 96, n. 2, p. 203–216, 2009. (Seção 6)

[27] VENCESLAU, A. W. do N. **Curvas Parametrizadas, Cicloides, Experimentos e Aplicações**. 2015. (Seção 3.4)

[28] WEISSTEIN, E. W. Ellipse. **From MathWorld–A Wolfram Web Resource**, 2022. Disponível em: https://mathworld.wolfram.com/Ellipse.html. Acesso 18 abr. 2022. (Seção 3.3, 3.4)

[29] WEISSTEIN, E. W. Cardioid. **From MathWorld–A Wolfram Web Resource**, 2022. Disponível em: https://mathworld.wolfram.com/Cardioid.html. Acesso 18 abr. 2022. (Seção 3.4)

[30] WEISSTEIN, E. W. Deltoid. **From MathWorld–A Wolfram Web Resource**, 2022. Disponível em: https://mathworld.wolfram.com/Deltoid.html. Acesso 29 abr. 2022. (Seção 3.5)

[31] WEISSTEIN, E. W. Astroid. **From MathWorld–A Wolfram Web Resource**, 2022. Disponível em: https://mathworld.wolfram.com/Astroid.html. Acesso 29 abr. 2022. (Seção 3.6)

[32] WEISSTEIN, E. W. Lemniscate. **From MathWorld–A Wolfram Web Resource**, 2022. Disponível em: https://mathworld.wolfram.com/Lemniscate.html. Acesso 27 out. 2022. (Seção 3.7)

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.

[33] WEISSTEIN, E. W. Eight curve. **From MathWorld–A Wolfram Web Resource**, 2023. Disponível em: https://mathworld.wolfram.com/EightCurve.html. Acesso 30 mar. 2023. (Seção 3.8)

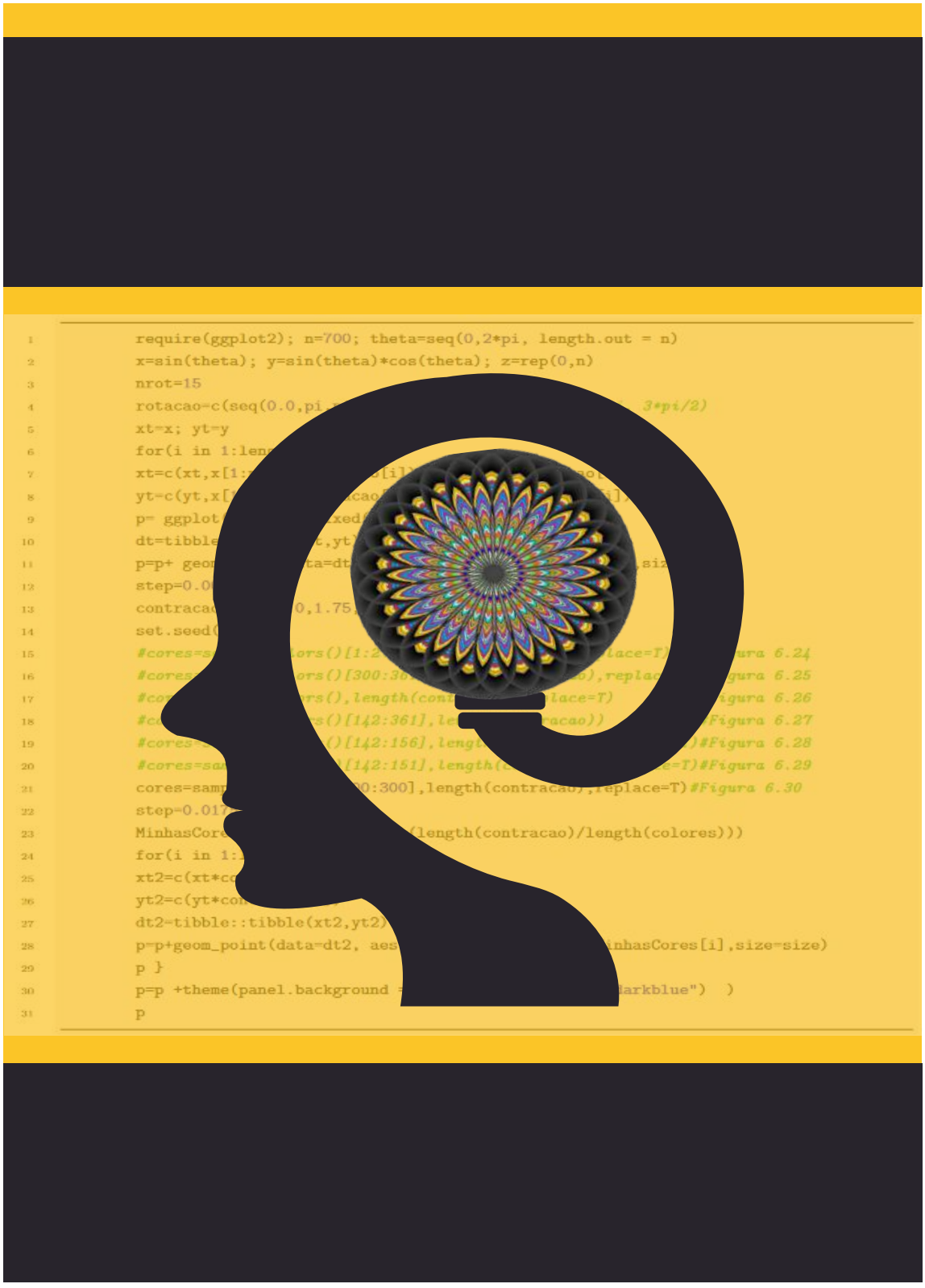
[34] WEISSTEIN, E. W. Limaçon. **From MathWorld–A Wolfram Web Resource**, 2023. Disponível em: https://mathworld.wolfram.com/Limacon.htmll. Acesso 30 mar. 2023. (Seção 3.9)

[35] WICKHAM, H. **ggplot2: Elegant Graphics for Data Analysis**. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. Disponível em: https://ggplot2.tidyverse.org. (Seção 1, 2.2)

[36] ZEILEIS, A. et al. colorspace: A toolbox for manipulating and assessing colors and palettes. **Journal of Statistical Software**, v. 96, n. 1, p. 1–49, 2020. (Seção 6)

[37] ZEILEIS, A.; HORNIK, K.; MURRELL, P. Escaping RGBland: Selecting colors for statistical graphics. **Computational Statistics & Data Analysis**, v. 53, n. 9, p. 3259–3270, 2009. (Seção 6)

ALCOFORADO, L.F.; SANTOS, J.P.M.; LIMA, M.V. ; FIRMIANO, A.; LÓPEZ LINARES, J. **Mandalas, curvas clássicas e visualização com R** Portal de Livros Abertos da USP, Pirassununga: Faculdade de Zootecnia e Engenharia de Alimentos, 2023. 105 p. ISBN 978-65-87023-33-5 (e-book). Disponível em: https://doi.org/10.11606/9786587023335.



ISBN 978-65-87023-33-5 (e-book)