

Fundamentos de Ecologia Numérica

Disciplina de Ecologia Numérica do Curso de Ciências Biológicas
do Campus V da UEPB

Prof. Elvio S. F. Medeiros Laboratório de Ecologia
Universidade Estadual da Paraíba
Campus V, João Pessoa, PB

Last updated: 09/02/2026 12:52:13

Contents

1 Apresentação	9
1.1 Motivação	9
1.2 Organização do livro	10
1.3 Como estudar por esse livro?	10
2 Introdução ao R/RStudio	11
2.1 Plano de estudo	12
2.2 Pré-requisitos	13
2.3 Principais conceitos	13
2.4 O que é o R Studio?	15
2.5 Instalação do R e RStudio	16
2.6 O essencial	17
2.7 Preparação dos arquivos	20
2.8 Conclusão	23
2.9 Sequência típica na preparação e análise de dados	23
2.10 Material de apoio	24
Apêndices	24
Sites consultados	24
Leitura suplementar	24
Referências	24
I BASES DE DADOS	25
3 Bases de dados. Dados do PPBio Semiárido	27
3.1 Sobre os dados do PPBio	27
3.2 Arquivos disponíveis	31
Referências	34
4 Base de dados I - Peixes	37
4.1 Introdução	38
4.2 Codificação das variáveis	41
4.3 Amostragem dos dados	43

4.4	Análises estatísticas	47
4.5	Discussão	47
	Apêndices	50
	Referências	55
5	Base de dados II - Habitat	57
	RESUMO	58
5.1	Introdução	59
5.2	Codificação para variáveis	61
5.3	Área de estudo e amostragem dos dados	64
5.4	Análises estatísticas	65
5.5	Resultados	65
5.6	Discussão	71
	Referências	73
	II ESTATÍSTICAS DESCRIPTIVAS	75
6	R Modulo 1 - Importando planilhas de dados	77
6.1	Sobre os dados do PPBio	78
6.2	Importando a planilha de trabalho	80
6.3	Mensagens de erro e avisos no R	81
6.4	Organização básica	82
6.5	Importando a planilha	84
6.6	Importando .ods do LibreOffice Calc	86
	Apêndices	90
	Script limpo	90
	Referências	91
7	R Modulo 2 - Tipos de dados no R	93
7.1	Tipos e estruturas de dados no R	94
7.2	Mensagens de erro e avisos no R	94
7.3	O que é um <code>script</code> e qual sua diferença para um <code>código</code> ?	95
7.4	Tipos e estruturas de dados no R	97
7.5	Tipos de dados no R	98
7.6	Estruturas de dados no R	100
7.7	Como pedir ajuda?	116
7.8	Work flow	120
7.9	Reconhecendo a estrutura dos dados do R em uma matriz multi-variada	120
7.10	Importando a planilha	121
	Apêndices	124
	Sites consultados	125
	Script limpo	125
	Referências	127

8 R Modulo 3.1 - Estatísticas descritivas e normalidade	129
8.1 Sobre os dados	130
8.2 Organização básica	130
8.3 Importando a planilha	132
8.4 Gráficos de histograma e boxplot	136
8.5 Sumário estatístico geral	137
Apêndices	138
Sites consultados	138
Script limpo	138
Referências	139
9 R Modulo 3.2 - Testes estatísticos	141
9.1 Sobre os dados	141
9.2 Organização básica	142
9.3 Importando a planilha	144
9.4 Gráficos de histograma e boxplot	147
9.5 Sumário estatístico geral	148
9.6 Testando normalidade	149
9.7 Testando diferenças entre médias	154
9.8 Correlação e regressão simples	172
Apêndices	180
Sites consultados	180
Script limpo	180
Referências	180
10 R Modulo 3.3 - Gráficos univariados	181
10.1 Sobre os dados	181
10.2 Organização básica	181
10.3 Importando a planilha	183
10.4 Reset point	185
10.5 Gráficos de dispersão	185
10.6 Gráficos de linha	187
10.7 Gráficos de barra	188
10.8 Gráficos de grupos	191
Apêndices	196
Sites para consulta	196
Script limpo	196
Referências	196
11 R Modulo 4.1 - Estrutura da comunidade	197
11.1 Sobre os dados	197
11.2 Organização básica	198
11.3 Importando a planilha	199
11.4 Reset point	200
11.5 Transpor a matriz para trabalhar com as espécies	201
11.6 Calculando os descritores da comunidade	206

11.7 Índices de Diversidade	212
11.8 Gráficos descritivos	232
11.9 Distribuição de abundância	235
11.10 Curva de rarefação	235
11.11 Curva de acumulação de espécies	239
11.12 Plot indivíduos	243
Apêndices	244
Sites consultados	257
Script limpo	257
Referências	265
12 R Modulo 4.2 - Decomposição da diversidade	267
12.1 Sobre os dados	267
12.2 Organização básica	268
Apêndices	268
Referências	268
13 R Modulo 4.3 - Estrutura do habitat	269
13.1 Sobre os dados	269
13.2 Organização básica	269
13.3 Importando a planilha	270
13.4 Reset point	271
13.5 Correlograma e remoção de variáveis redundantes ou desnecessárias	272
13.6 Deletando variáveis colineares	277
13.7 Somando variáveis redundantes	278
13.8 Tabela de dados ambientais	281
Apêndices	284
Sites consultados	284
Script limpo	284
Referências	284
14 R Modulo 5 - Relativizações e transformações	285
14.1 Organização básica	285
14.2 Tamanho da matriz	288
14.3 Reset point 1	290
14.4 Cálculo da matriz de distâncias	291
14.5 Distribuição de frequência da matriz de distâncias	294
14.6 Distribuição de frequências da matriz de distâncias	295
14.7 Relativizando e transformando a base de dados	296
14.8 Reset point 2	302
Apêndices	303
Script limpo	305
Referências	308
15 R Módulo 6 - Análise de outliers baseada no desvio padrão do centróide	309

15.1 Sobre os dados do PPBio	309
15.2 Organização básica	311
15.3 Reset point	317
15.4 Cálculo de matriz de distâncias Euclidiana (<i>euclid</i>)	320
15.5 Calculando o centróide da matriz de distâncias <i>euclid</i>	323
15.6 Distribuição de frequências da matriz de distâncias <i>euclid</i>	324
15.7 Procurando outliers	326
15.8 Lista final de outliers	333
15.9 Particionando a matriz	336
15.10 Exportando a matrix final de trabalho	340
Apêndices	341
Script limpo	342
Referências	346
16 R Modulo 10-1	347
16.1 Algoritmo da NMDS	348
16.2 Organização básica	349
16.3 Pacotes do módulo	349
16.4 REINÍCIO 1	349
16.5 Classificação 1: Matriz comunitária	349
16.6 Ordenação	356
Apêndices	378
Sites consultados	378
Referências	379
III ANÁLISES MULTIVARIADAS	381
17 R Módulo 7 - Análise de Classificação Hierárquica - SAHN	383
17.1 Organização básica	384
17.2 Reset point	387
17.3 Nomenclatura das matrizes em AMD	387
17.4 Cluster 1: Piloto automático	387
17.5 Cluster 2: Criando a matriz de distâncias Euclidiana	397
17.6 Cluster 3: Relativização pelo total da coluna	400
17.7 Cluster 4: Transformação pelo arcoseno da raiz quadrada	401
17.8 Cluster 5: Controle total dos parâmetros	402
17.9 Construindo uma prancha de gráficos comparativos	407
17.10 Construindo <i>heatmaps</i>	408
Apêndices	415
Sites consultados	415
Script limpo	416
Referências	421

Chapter 1

Apresentação

1.1 Motivação

Este é um site experimental da disciplina de graduação de Ecologia Numérica do Curso de Ciências Biológicas da Universidade Estadual da Paraíba.

O site experimental da disciplina de Ecologia Numérica da UEPB é uma plataforma criada com o objetivo de apresentar exemplos didáticos aos alunos, explorando diferentes conceitos e técnicas utilizadas na análise de dados em Ecologia. É importante destacar que os exemplos apresentados no site podem conter imprecisões ou códigos demasiado extensos (aparentemente desnecessários para um ecólogo e programador de R mais experiente), mas isso é proposital, pois a finalidade é oferecer um material didático para os alunos, que possa facilitar a compreensão dos conceitos abordados durante as aulas. Dessa forma, o site é uma ferramenta complementar às aulas presenciais, que busca fornecer exemplos práticos e didáticos para os estudantes de Ecologia Numérica da UEPB.

Como mencionado, a finalidade é didática e de apresentar o tema seguindo passo-a-passo e contextualizando com dados ecológicos reais. Por isso, além das informações sobre o conteúdo do site experimental da disciplina de ecologia numérica da UEPB, é importante destacar que qualquer erro ou problema relacionado ao site pode ser reportado ao responsável pelo site.

É fundamental que os usuários do site compreendam que plágio e outras formas de má conduta acadêmica são inaceitáveis e podem ter sérias consequências. O responsável pelo site se compromete a remover qualquer conteúdo ofensivo ou sem citar a devida fonte consultada, que possa ter sido publicado erroneamente ou por engano.

1.2 Organização do livro

Este livro é organizado em CINCO partes:

Na PARTE I, são apresentadas as bases teóricas fundamentais sobre ecologia numérica e análise multivariada.

Na PARTE II, é apresentada a fundamentação teórica para o entendimento das BASES DE DADOS que são trabalhados ao longo das PARTES III e IV e execução das estatísticas descritas.

Na PARTE III, são apresentados os primeiros Módulos do RStudio sobre ESTATÍSTICAS DESCRIPTIVAS EM ECOLOGIA DE COMUNIDADES, na forma de tutoriais discutidos, que podem ser executados passo-a-passo sobre .

Na PARTE IV, são apresentados os Módulos do RStudio mais avançados, sobre TÉCNICAS DE ANÁLISE MULTIVARIADA EM ECOLOGIA DE COMUNIDADES.

Por fim, na PARTE V, são apresentadas informações gerais de apoio para o melhor entendimento do conteúdo, como uma lista geral das tabelas usadas na PARTE II, um glossário de termos, bibliografia geral, entre outras fontes de informação.

1.3 Como estudar por esse livro?

Simples! Siga a sequência que é apresentada.

- Estude a teoria (PARTE I), certifique-se de que entende os conceitos e é capaz de tomar decisões fundamentadas na teoria. Revise/relembre do Ensino Médio a matemática necessária (Conjuntos, Matriz, Álgebra, Teorema de Pitágoras, etc.).
- Leia a fundamentação das bases de dados que usará subsequentemente (PARTE II). Mas lembre, você não precisa ser especialista nos grupos taxonômicos apresentados aqui como exemplo. O ecólogo trabalha com dados, e eles falam por si só.
- Parta para a ação (PARTES III e IV), comece com os primeiros módulos do R, siga as instruções passo-a-passo, execute linha por linha de código, atente para os resultados.
- Consulte os detalhes e referências apresentados na PARTE V.

1.3.1 Para um estudo aprofundado

Chapter 2

Introdução ao R/RStudio

RESUMO

Bem-vindo à disciplina de Ecologia Numérica. A base desta disciplina está no uso de técnicas multivariadas de classificação e ordenação, como são usadas em ecologia e gestão de recursos. Todos os textos e aulas foram elaborados para serem complementados pela Bibliografia Básica e pelo Material de Apoio (ver subsequentemente). Usaremos o software RStudio!, desenvolvido por RStudio Team (2022) (<http://www.rstudio.com/>).

Embora seja necessário algum conhecimento matemático e uma base em bioestatística, estes assuntos não têm uma abordagem basicamente matemática e o material de apoio será usado para revisar temas de outras disciplinas. Os procedimentos são explicados em termos de fórmulas e algoritmos e o aluno obterá pelo menos uma compreensão intuitiva do que os métodos envolvem. Para aqueles com mais interesse nos temas tratados, a bibliografia básica oferece maior profundidade. A Ecologia Numérica é a ferramenta básica do biólogo no tocante à análise quantitativa de quaisquer tipos de dados. Ela cobre, além dos assuntos abordados nessa disciplina, áreas de rápido desenvolvimento de análise temporal e espacial. Esperamos que, ao concluir esta disciplina, o texto seja de valor duradouro em seu trabalho profissional.

Os softwares disponíveis para análise multivariada são diversos. A maioria dos pacotes maiores tem algumas das rotinas padrão, mas um software especializado, como o R/RStudio, também é necessário para fazer o melhor uso dos avanços recentes em aplicações ecológicas. O R é um ambiente e uma linguagem de programação gratuita e de “código aberto”. Veremos mais sobre isso subsequentemente.

Legendre e Legendre (1998) alertam o ecologista numérico sobre as armadilhas do uso de computadores - que a **computabilidade não garante (i) que os dados satisfaçam o método e (ii) que os resultados sejam inter-**

pretados corretamente em termos ecológicos. A isso, infelizmente, deve ser adicionada a advertência adicional de que os computadores não negam a necessidade de verificar a exatidão dos cálculos.

A conclusão desta disciplina ajudará a evitar essas armadilhas, fornecendo a você as habilidades para aplicar essas técnicas com sucesso para o avanço da ciência e do gerenciamento de nossos recursos naturais. Espero que com a conclusão dessa disciplina o aluno possa evitar essas armadilhas e possua as habilidades para aplicar essas técnicas com sucesso para avançar na sua área de atuação.

2.1 Plano de estudo

O objetivo principal é familiarizar-se com o uso do RStudio para que você possa executar as operações básicas com confiança. Você aprenderá como instalar o software, importar e exportar dados, exibi-los em forma de planilha, modificá-los das principais formas e exibir informações básicas. Além disso, você também aprenderá a exportar resultados para permitir a apresentação e análises posteriores em outros pacotes estatísticos. Ao ler os artigos de revisão em Leitura Suplementar, você será apresentado a algumas das diversidades do trabalho multivariado e as críticas que têm recebido na ecologia, mas também um excelente exemplo de uma análise multivariada cuidadosamente descrita. Ele leva você através da configuração objetiva, seleção de caracteres e codificação de dados, em seguida, em várias análises e, em seguida, o desenho de conclusões apropriadas.

Portanto, para obter o máximo benefício, recomenda-se que você faça o seguinte:

- leia a o conteúdo Teórico inteiro sobre conceitos-chave minuciosamente;
- leia e estude todo o Material de Apoio disponível do Google Sala de Aula;
- complete os exemplos e tutorias passo-a-passo (quando houver);
- leia os artigos introdutórios na Leitura Suplementar (esse passo não é obrigatório nem necessário para completar o curso);
- participe e tire dúvidas no Chat da disciplina no Google Sala de Aula;
- complete os Exemplos aplicados;
- avalie sua compreensão dos materiais, usando os quizzes, que são perguntas de múltipla escolha baseadas no Google Sala de Aula.

Conforme você avança ao longo da disciplina, concentre-se mais no fluxo de procedimentos para entrada (input) e modificação de dados, em vez de nos detalhes das opções e saídas (outputs). Eles sempre podem ser consultados quando necessário, aqui ou no sistema de ajuda do R/RStudio.

Os scripts do R/RStudio documentam o histórico de sua análise. Também é aconselhável, ao fazer uma análise, registrar notas adicionais sobre a interpretação da análise (usando `#`). Isso também pode ser feito diretamente nos arquivos de resultados.

Ao completar cada Módulo o aluno deve ser capaz de:

instalar o R e RStudio
decidir a melhor forma de preparar seus dados em vários formatos (.xlsx, .csv, etc)
configurar diretórios de projetos
importar, modificar e exportar seus dados
preparar um arquivo de nome de espécies
particionar arquivos de dados em subconjuntos
considerar o tratamento de valores faltantes
exportar resultados para outros softwares para análise e apresentação
salvar e documentar um arquivo de resultados

Ao estudar cada módulo, concentre-se mais no fluxo dos procedimentos para entrada e modificação de dados, em vez dos detalhes das opções. Estes podem sempre ser revisados quando necessário, aqui ou nos materiais de apoio. A janela do Console ou (“Work Area”) documenta o histórico de sua análise. Também é aconselhável ao fazer uma análise, registrar e fazer anotações adicionais sobre sua interpretação da análise etc. Isso pode ser feito diretamente na janela de Scripts.

2.2 Pré-requisitos

Familiaridade básica com computadores, tipos de arquivos (.xlsx, .docx, etc), processadores de texto e planilhas é necessária aqui. Espera-se que você seja capaz de compreender estatísticas univariadas básicas.

2.3 Principais conceitos

2.3.1 O ambiente de programação do R

O programa R oferece uma poderosa ferramenta para manipulação, análise e visualização de dados, e nos últimos anos tornou-se muito popular entre os ecólogistas (e não apenas eles). Ele oferece grande liberdade e permite que você faça o que quiser com seus dados, o que não é o caso de softwares clicáveis como SAS, SPSS ou STATISTICA, entre outros. No entanto, ele também surge com uma curva de aprendizado íngreme, tendo em vista sua linguagem baseada em S/S-Plus, e um certo nível de frustração por conta de mensagens de erro frequentes. A menor imprecisão na digitação do código impede que ele seja executado, e gasto de tempo na busca do erro. Isso, em parte, estimula a “copia-cola” de scripts prontos, levando o usuário a executar análises sem saber o que está fazendo. Exceto pelo livro de Jean Valentin (VALENTIN, 2000), a grande maioria dos textos impressos em ecologia numérica são em inglês. Assim como inúmeros tutoriais em inglês pela internet (veja Introduction to R for Ecologists de David Zelený. Em português temos o excelente livro on-line Análises Ecológicas no R de SILVA et al. (2022).

O R não é um software do tipo aplicativo, é um ambiente de programação com

um conjunto integrado de ferramentas de software para manipulação de dados, cálculo e apresentação gráfica. A linguagem R foi elaborada inicialmente com o objetivo específico de análises de dados, na qual muitas técnicas estatísticas, clássicas e modernas, podem ser implementadas, entre outras coisas não tão estatísticas. Algumas dessas técnicas estão implementadas no ambiente básico do R (“R base”), mas muitas estão implementadas em pacotes adicionais (“packages”).

O ambiente R foi desenvolvido baseado na linguagem S, no final da década de 90 início dos anos 2000. A sua estrutura de código aberto (que vem da linguagem S) e de software público e gratuito atraiu um grande número de desenvolvedores, sendo que há hoje inúmeros pacotes para o R (veja Introdução ao R para Análise de Dados em Consultoria Ambiental).

Para um excelente minicurso introdutório sobre o R clique [aqui](#)

“Uma das coisas mais importantes que você pode fazer é dedicar um tempo para aprender uma linguagem de programação de verdade [e aprender a base teórica e estatística]. Aprender a programar é como aprender outro idioma: exige tempo e treinamento, e não há resultados práticos imediatos. Mas se você supera essa primeira subida íngreme da curva de aprendizado, os ganhos como cientista são enormes. Programar não vai apenas livrar você da camisa de força dos pacotes estatísticos, mas também irá aguçar suas habilidades analíticas e ampliar os horizontes de modelagem ecológica e estatística.”

— (Nicholas J. Gotelli em Gotelli & Ellison 2011)

Os softwares disponíveis para análise multivariada são diversificados. A maioria dos pacotes tem algumas das análises padrão, mas o R é necessário para fazer o melhor uso dos recentes avanços em aplicações ecológicas. A literatura, por outro lado, alerta o ecologista numérico para as armadilhas do uso de computadores. A computação não garante (i) que os dados satisfaçam o método, e (ii) que os resultados sejam interpretados corretamente em termos ecológicos. Para isso, é necessária uma base teórica forte além de que os computadores não negam a necessidade de verificar a correção dos cálculos. Uma das ferramentas mais populares para análise de dados no R é o RStudio, uma interface de desenvolvimento integrada (IDE) que oferece recursos avançados para desenvolvimento, depuração e gerenciamento de projetos em R. Embora o RStudio seja uma ferramenta poderosa para análise de dados, é importante lembrar que a análise multivariada não depende apenas do software utilizado, mas também de uma base teórica forte e de verificação da correção dos cálculos. Além disso, a literatura alerta que o uso de computadores não garante a adequação dos dados ao método e a interpretação correta dos resultados em termos ecológicos. Portanto, é importante combinar o uso de ferramentas como o RStudio com uma sólida formação teórica e verificação rigorosa dos resultados.

2.4 O que é o R Studio?

O RStudio é um ambiente integrado de desenvolvimento (IDE) para a linguagem de programação R, projetado para ajudar os usuários a trabalhar com dados de forma mais eficiente e produtiva (Figura 2.1). Ele oferece uma ampla variedade de recursos e ferramentas para análise de dados, incluindo edição de código, visualização de gráficos, depuração, gerenciamento de projetos e colaboração. O RStudio é gratuito e de código aberto, disponível para Windows, Mac e Linux, e é usado por uma comunidade crescente de cientistas de dados, pesquisadores, analistas e desenvolvedores em todo o mundo. Ele oferece suporte para várias bibliotecas e pacotes do R, incluindo o `tidyverse`, que é uma coleção popular de pacotes para manipulação de dados e visualização. Com o RStudio, os usuários podem facilmente importar, manipular e visualizar dados, executar análises estatísticas e criar visualizações gráficas interativas. Além disso, o RStudio oferece a possibilidade de publicar relatórios e apresentações dinâmicas em diversos formatos, como HTML, PDF e Word, facilitando a comunicação e colaboração entre os usuários.

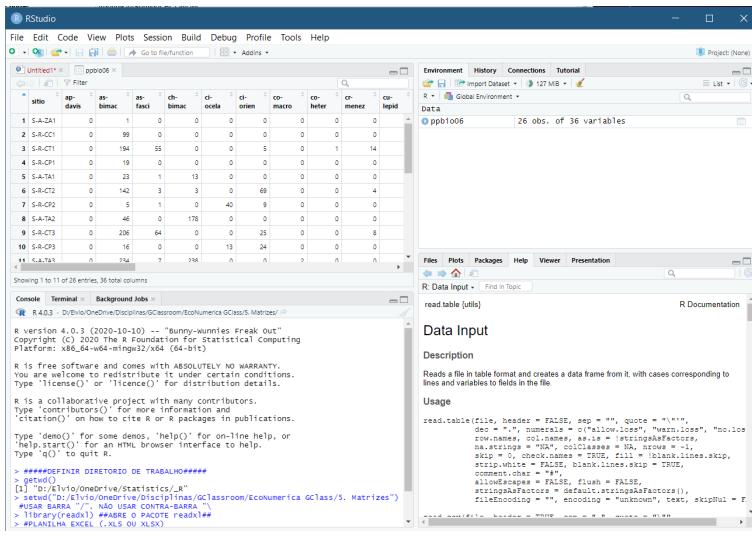


Figure 2.1: Interface do usuário do ambiente integrado de desenvolvimento (IDE) RStudio para a linguagem de programação R.

O RStudio oferece diversas vantagens em relação ao uso do R base. Algumas dessas vantagens incluem:

1. Interface amigável: o RStudio possui uma interface gráfica amigável e intuitiva, com recursos como realce de sintaxe, sugestões de código e facilidade na navegação entre arquivos, o que torna o processo de desenvolvimento mais fácil e produtivo.
2. Facilidade de gerenciamento de projetos: o RStudio oferece ferramentas

para gerenciamento de projetos, que permitem aos usuários organizar e controlar arquivos e scripts em um ambiente centralizado.

3. Depuração eficiente: o RStudio oferece recursos de depuração integrados, como pontos de interrupção e rastreamento de variáveis, que facilitam a identificação e correção de erros no código.
4. Suporte para colaboração: o RStudio permite que múltiplos usuários colaborem em um projeto, com recursos como controle de versão integrado e ferramentas para compartilhamento de código.
5. Integração com outras ferramentas: o RStudio integra-se com diversas ferramentas populares de análise de dados, como o Git e o LaTeX, permitindo uma maior eficiência e produtividade no processo de análise e geração de relatórios.

Essas são apenas algumas das vantagens que o RStudio oferece em relação ao R. É importante notar que, embora o RStudio seja uma ferramenta poderosa para análise de dados, ele não substitui a necessidade de conhecimento em R e estatística para obter resultados precisos e significativos.

2.5 Instalação do R e RStudio

2.5.1 R base

1. O primeiro passo é entrar na página do projeto CRAN (Comprehensive R Archive Network) (<https://www.r-project.org/>).
2. Do lado esquerdo da página clique sobre o link CRAN abaixo de Download. 3. Uma nova página com uma série de links irá se abrir. Esses links são chamados de “espelhos” e servem para que você possa escolher o local mais próximo de onde você está para fazer o download do programa. Escolha um espelho no Brasil.
3. Na seção Download and Install R, clique sobre o link **Download R for Windows** para baixar a versão para esse sistema... (MacOS??... o que é isso?)
4. Clique sobre o link **base**.
5. Clique sobre o link **Download R 4.x.x for Windows** para fazer o download do arquivo R.exe.
6. A instalação segue o formato padrão de instalação de programas no Windows, e portanto não são necessários maiores detalhes.

2.5.2 RStudio

1. Para baixar o RStudio entre no endereço (<https://posit.co/download/rstudio-desktop/>)

2. Clique no link **Products > RStudio**
3. Selecione a versão Desktop.
4. Clique em **DOWNLOAD RSTUDIO DESKTOP**
5. Será exibida uma página com a recomendação para você baixar o RStudio FREE versão mais recente - Windows
6. Clicando nesse link, você irá baixar o arquivo RStudio atual (.exe)
7. Depois é só clicar e instalar da forma convencional do Windows.

Após a instalação, você pode abrir o **RStudio** pelo seu respectivo ícone, e o RStudio estará pronto para ser utilizado. O “R base” continuará instalado mas será acessado pelo RStudio. Não o desinstale.

2.5.3 RStudio na nuvem

Usar o RStudio Cloud (<https://login.rstudio.cloud/>) é uma opção para quem não quer instalar a versão para PC. O RStudio Cloud é uma plataforma online que fornece um ambiente de desenvolvimento integrado para o R, permitindo que os usuários executem análises, desenvolvam código e colaborem com outras pessoas, sem a necessidade de instalar o R e o RStudio em seus próprios computadores. É uma solução conveniente e acessível, especialmente para iniciantes ou usuários que desejam compartilhar projetos e colaborar de forma eficiente.

2.6 O essencial

2.6.1 Licença de software

A licença de uso do RStudio depende da versão do software que está sendo utilizada. O RStudio oferece duas versões principais: a versão gratuita e a versão paga, conhecida como RStudio Pro. A versão gratuita do RStudio é licenciada sob a licença GPLv3, o que significa que é um software de código aberto e livremente distribuível. Isso permite que os usuários baixem, modifiquem e distribuam o código-fonte do RStudio gratuitamente. No entanto, o uso comercial do RStudio sob a licença GPLv3 pode ter algumas limitações, como a obrigatoriedade de disponibilizar o código-fonte modificado para os usuários. Já a versão paga do RStudio, o RStudio Pro, requer uma licença comercial para uso em ambientes empresariais ou acadêmicos. A licença comercial do RStudio Pro oferece recursos adicionais, como suporte técnico e atualizações regulares do software. Além disso, a licença comercial permite que os usuários utilizem o RStudio Pro em um ambiente sem a obrigação de divulgar o código-fonte modificado. É importante lembrar que as licenças de uso do RStudio podem mudar com o tempo, portanto, é recomendado que os usuários verifiquem as informações de licenciamento diretamente no site do RStudio antes de utilizar

o software.

2.6.2 Interface do usuário

Uma interface de programa estatístico é uma ferramenta que permite que os usuários interajam com uma linguagem de programação de maneira mais amigável e visual. Essas interfaces geralmente apresentam uma interface gráfica do usuário (GUI) que oferece um ambiente integrado para a análise de dados. Os usuários podem importar dados, criar visualizações, executar análises estatísticas e gerar relatórios. As interfaces de programa estatístico podem variar desde aplicativos de desktop autônomos até plug-ins para aplicativos de planilha, e muitos são desenvolvidos para atender a necessidades específicas de usuários em áreas como finanças, pesquisa médica, ciências sociais e outras. Essas interfaces tornam a análise de dados mais acessível a uma gama mais ampla de usuários, independentemente de sua experiência em programação ou estatística.

Linguagens de programação estatística de código aberto tem sido amplamente usadas em análises de dados e pesquisa em todo o mundo. Suas interfaces ou GUIs, embora poderosas, podem ser um pouco intimidadoras para novos usuários, uma vez que é necessário habilidades de programação básicas.

2.6.3 Interface do usuário do RStudio

A interface de usuário do RStudio é composta por várias janelas e painéis que permitem a interação com o R de forma organizada e intuitiva. A janela principal é dividida em quatro painéis (Figura 2.2):

Editor de código: é onde o usuário pode criar, editar e salvar scripts R. O editor fornece recursos como destaque de sintaxe, indentação automática, autocompletar e outras ferramentas úteis para a programação.

Console: é onde o usuário pode digitar comandos diretamente no R e ver a saída correspondente. É aqui que a maior parte da interação com o R ocorre.

Arquivos/Plots/Pacotes/Ajuda: este painel pode ser dividido em quatro subpainéis, cada um dos quais mostra uma guia diferente. O painel de arquivos permite a navegação pelos arquivos e pastas do computador. O painel de plots mostra gráficos gerados no R. O painel de pacotes permite a instalação, atualização e carregamento de pacotes do R. O painel de ajuda exibe informações sobre funções e pacotes do R.

Ambiente/Histórico/Git/Conexões: este painel também pode ser dividido em quatro subpainéis. O painel de ambiente exibe informações sobre variáveis e funções criadas no R. O painel de histórico mostra os comandos recentes digitados no console. O painel de Git permite que o usuário integre o RStudio com um repositório Git para controle de versão. O painel de conexões permite a conexão com bancos de dados e outros recursos externos.

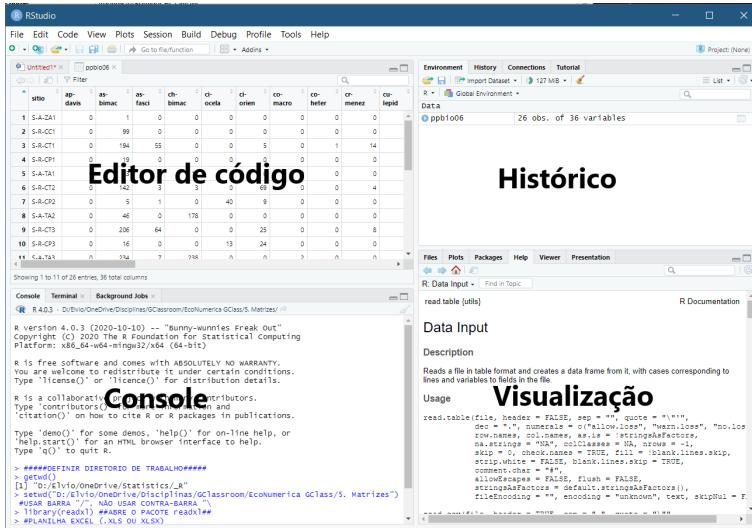


Figure 2.2: Interface do usuário do ambiente integrado de desenvolvimento (IDE) RStudio para a linguagem de programação R com os termos em inglês e em português. O operador de atribuição (`<-`), um dos operadores mais utilizados no R pode ser chamado pela combinação alt + - (alt e a tecla menos).

A interface de usuário do RStudio também inclui uma barra de ferramentas na parte superior da janela, que fornece acesso rápido a comandos e recursos comuns. Além disso, há uma barra de menus padrão que permite o acesso a todas as opções e recursos do RStudio. No geral, a interface de usuário do RStudio é altamente personalizável e pode ser configurada para atender às necessidades específicas do usuário.

2.6.4 Projetos

Um ‘Projeto’ no RStudio é um conjunto de dados e arquivos de resultados associados entre si, junto com as opções e configurações no momento em que o Projeto foi salvo pela última vez. Quando você abre um projeto, esses arquivos são abertos e suas opções e configurações são restauradas da última vez. Você só pode ter um projeto aberto por vez, mas pode alternar facilmente entre eles abrindo um projeto salvo. Um projeto pode ter mais de um arquivo de dados associado. Se você tiver um projeto aberto e abrir ou criar qualquer arquivo nele, o projeto será considerado modificado. Ao sair do RStudio ou abrir um novo projeto, será perguntado se deseja salvar o projeto modificado. Para melhor gerenciamento de dados, você pode manter seus arquivos de projeto em seus próprios diretórios, separados do software RStudio. Ele lembra o nome do caminho completo para cada projeto. Os detalhes de como o RStudio gerencia diretórios e definições de gerenciamento de arquivos são fornecidos no sistema de Ajuda. Você pode redefinir o diretório padrão (working directory) para que o

RStudio navegue até o diretório que você escolheu e abra os arquivos ou projetos que você encontrará no diretório escolhido.

2.6.5 Nomes de arquivos

Alguns dos tipos de extensão de arquivos que podem ser criados ou manipulados no RStudio incluem:

Arquivos de script: São arquivos de texto contendo código R, com extensões de arquivo como .R, .Rmd, .Rnw, .Rhistory, entre outros.

Arquivos de dados: São arquivos contendo dados que podem ser lidos e manipulados com o R, como .csv, .txt, .xls, .xlsx, .rda, .rdata, entre outros.

Arquivos de pacotes: São arquivos compactados que contêm código, documentação e dados necessários para um pacote do R, com extensão .tar.gz.

Arquivos de imagem: São arquivos de imagem gerados a partir de gráficos criados no R, com extensões como .pdf, .png, .jpeg, .bmp, entre outros.

Arquivos de projeto: São arquivos que contêm todas as informações necessárias para reproduzir um projeto R, incluindo arquivos de script, dados e outros recursos, com extensão .Rproj.

Esses são apenas alguns dos tipos de arquivos que podem ser usados ou gerados no RStudio. O RStudio também oferece suporte a muitos outros tipos de arquivos, como arquivos de ajuda, arquivos de instalação de pacotes, entre outros.

2.7 Preparação dos arquivos

Arquivos de dados multivariados são geralmente maiores e mais complexos do que outros, portanto, é necessário cuidado extra. Eles podem ser preparados de duas maneiras principais. A primeira forma é como a matriz completa de valores (.xlsx), onde cada linha mostra os valores de cada coluna. Isso é bom quando há poucos zeros. É assim que uma planilha funciona. A outra maneira, preferida pelos ecologistas para seus dados de espécies, é registrar apenas os valores diferentes de zero para cada espécie. Cada linha, representando uma unidade de amostragem, é então lida como uma lista de verificação ou registro quadrat. Este formato compacto (.csv ou tab delimited) é muito mais eficiente quando o tamanho da matriz de dados é muito grande. O RStudio considera as linhas no arquivo de dados como as unidades de amostragem e as colunas como as variáveis. Dependendo de quantos existem, pode ser mais conveniente inserir seus dados ao contrário e, em seguida, transpô-los para o RStudio. Veremos como fazer isso mais tarde.

2.7.1 Formatos de arquivo de dados

Você pode usar qualquer editor de texto, processador de texto ou planilha para construir sua planilha de dados, embora seja recomendado o uso do Excel. No

entanto, a entrada de dados no RStudio deve ser feita dentro do ambiente R por linha de comando. Os dados podem ser preparados no formato de planilha (spreadsheet), mas você deve seguir o formato cuidadosamente para saber onde e como colocar as informações nas células da planilha.

A seguir é apresentada a estrutura básica de uma matriz comunitária e como importar um arquivo XLS/XLSX para o RStudio usando a programação R (Figura 2.3). Como tudo no R, existem várias formas para se ler os arquivos Excel. Usaremos primeiro pacote `readxl` (que já vem instalado com o R base) (veja <https://www.tutorialkart.com/r-tutorial/r-read-excel-xls-xlsx/>)

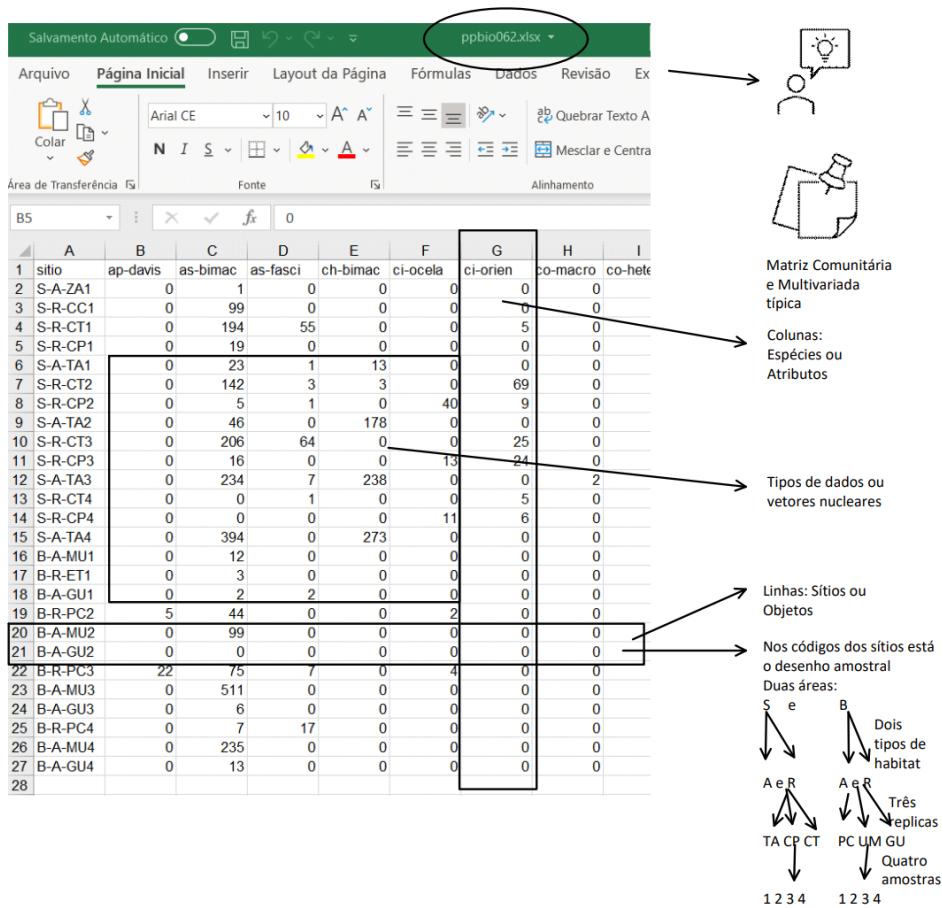


Figure 2.3: Usaremos uma matriz multivariada (sítios x espécies, matriz comunitária) do PPBio. Identifique na figura abaixo os elementos que compõem uma matriz multivariada

2.7.2 O arquivo de espécies

Ao usar o formato de dados compacto, você viu que as variáveis (colunas) são numeradas, não nomeadas. Portanto, para aplicar um nome a eles, você precisa criar um arquivo de texto separado que contenha os códigos e nomes. Isso é mais útil quando as variáveis são espécies, uma vez que você pode ter um único arquivo de espécie que se aplica a todos os seus arquivos de dados e aos projetos.

O arquivo de espécies também inclui o nome completo da espécie, que é usado em alguns arquivos de resultados. Esta facilidade é obviamente muito útil para registro de campo quando você pode identificar a maioria das espécies. Leve a lista de espécies com você e apenas escreva o número do código em sua folha de dados. Observe também os seguintes guias de formato e regras para o arquivo de espécies.

Nomes de código curtos são recomendados para reduzir a superposição de nomes em quaisquer resultados gráficos.

Os registros podem estar em qualquer ordem neste arquivo. (No entanto, você achará útil manter as espécies em ordem alfabética porque a maior parte dos outputs é ordenada pela sequência de espécies em seu arquivo de espécies.).

2.7.3 Formato de valores separados por vírgula (.csv)

Um arquivo ASCII de valores separados por vírgula (formato *.csv) também pode ser usado para importar / exportar dados do RStudio. Este formato de arquivo requer uma formatação muito cuidadosa.

2.7.4 Editando a matriz de dados

O RStudio fornece uma plethora de formas para edição de matrizes. Isso é mais útil para pequenas correções. Essas edições são salvas no formato de planilha para que você possa alternar entre a edição do arquivo e o software de planilha. Use a planilha para uma edição mais extensa de suas matrizes. Você pode fazer operações do tipo matriz facilmente na janela principal. A exclusão de linhas ou colunas, transformações e transposição e multiplicação de matrizes estão disponíveis com linhas de comando.

2.7.5 Arquivos de resultados

Para manter um registro completo de suas análises, é melhor anexar arquivos de resultados sucessivos a um arquivo de scripts nomeado. Faça isso salvando o arquivo de resultado e, então, após cada operação sucessiva, selecione Salvar script. Você também deve adquirir o hábito de inserir um título descritivo significativo (usando #) para cada análise. Isso fornece documentação adicional de sua análise e aparece no script. As tabelas no arquivo de resultado são construídas com espaços, não tabulações. Portanto, o alinhamento das colunas no RStudio e nos processadores de texto só é obtido selecionando uma fonte de

pitch-fixo, como Courier. Sendo arquivos de texto simples, você também pode pesquisar ou editar arquivos de resultados e scripts usando o R ou qualquer editor de texto.

Você pode exportar seus arquivos de resultado para sua planilha.

2.7.6 Gráficos

Você pode salvar imagens gráficas do RStudio como metarquivos do Windows (.wmf) ou metarquivos aprimorados (*.emf), usando linhas de comando ou a interface gráfica. Um gráfico salvo dessa forma é uma imagem vetorial que pode ser aberta em um pacote gráfico ou colada em processadores de texto e redimensionada ou editada sem perda de detalhes. No entanto, certifique-se de não alterar a proporção de aspecto ao redimensionar. Se o formato *.emf não for lido corretamente por seu outro software, tente salvar o gráfico no formato .wmf.

2.8 Conclusão

Foram apresentadas as três primeiras etapas da sequência abaixo, incluindo a preparação da importantíssima matriz de dados e como importá-la e exportá-la. Resumir os dados e decidir sobre o tratamento dos valores ausentes é um primeiro passo para a familiarização com conjuntos de dados multivariados.

2.9 Sequência típica na preparação e análise de dados

Para realizar uma análise multivariada, precisamos seguir uma sequência de etapas analíticas, como a seguir:

1. Entrada de dados e verificação de erros
Crie os arquivos de dados e espécies usando sua planilha ou editor de texto ou banco de dados. Verifique e arquive uma cópia do seu arquivo de dados limpo.
2. Crie o arquivo de planilha .xlsx
Faça isso salvando a planilha ou importando outros formatos para o RStudio. Verifique se há erros de importação. Faça resumos para reproduzir seus registros de unidade de amostragem para dupla verificação.
3. Examine a estrutura básica dos dados
Use os comandos de sumários para examinar os valores de linha e coluna
4. Faça uma Análise de Outliers
Procure outliers. Examine as relações entre curvas de espécies e espécie-área.

5. Decida se/e como modificar os dados
Mesclar, particionar, transformar, podar, transpor matriz etc.
6. Fazer análises, ordenar e classificar (grupos)
Cada análise pode exigir o carregamento de várias matrizes principais e secundárias
7. Ispencionar os resultados (gráficos e arquivos de resultados)
Salve e imprima os resultados

2.10 Material de apoio

Recursos na Web Apostilas RStudio Desktop - Posit The Comprehensive R Archive Network (r-project.org) Landeiro-Introducao.pdf (r-project.org)

Apêndices

Sites consultados

Como importar dados do Excel para o R: [<https://youtu.be/U6ksXvvY6Q0>]
Como exportar dados do R para o Excel: [https://youtu.be/a7EJE_2mtGk]

Leitura suplementar

A multivariate analysis on spatiotemporal evolution of Covid-19 in Brazil (<https://doi.org/10.1016/j.idm.2020.08.012>)
Genetic markers in the playground of multivariate analysis (<https://www.nature.com/articles/hdy2008130>)

Referências

Part I

BASES DE DADOS

Chapter 3

Bases de dados. Dados do PPBio Semiárido

Exemplos de conjunto de dados ecológicos do Programa de Pesquisa em Biodiversidade (PPBio) do Semiárido

RESUMO

Para entender a distribuição das espécies de peixes e seu uso de habitat, uma série de variáveis ambientais foram avaliadas como preditores da composição e riqueza da assembleia de peixes em sistemas aquáticos tropicais semiáridos. Nós pesquisamos a composição de espécies de assembleias de peixes em sistemas aquáticos semiáridos e estabelecemos seu grau de associação com a estrutura do habitat aquático. Os locais consistiam em trechos de riachos com fluxo de água superficial, poças temporárias isoladas e reservatórios artificiais (açudes). A amostragem de peixes foi realizada em quatro ocasiões durante as estações chuvosa (abril e junho de 2006) e seca (setembro e dezembro de 2006).

Palavras-chave: rios intermitentes, reservatórios, conservação, composição de substratos.

3.1 Sobre os dados do PPBio

Usaremos ao longo desse livro dados que fazem parte de um estudo mais amplo sobre ecologia de rios do semiárido, coletados no Programa de Pesquisa em Biodiversidade - PPBio (Veja Programa de Pesquisa em Biodiversidade – PPBio).

Nesta base de dados estão armazenadas informações sobre diversos grupos taxonômicos distribuídos em diversas unidades amostrais (UA's ou sítios), como peixes, macroinvertebrados bentônicos, quironomídeos e zooplâncton, além de

dados do habitat, como variáveis físicas e químicas, morfologia do habitat, composição do substrato, estrutura de habitat marginal, entre outros (veja a seção Arquivos disponíveis, a seguir).

Parte desses dados está armazenada em planilhas de Excel `ppbio**.xlsx` (Figura 3.1). Essas matrizes de dados são descritas na Tabela ???. As planilhas `ppbio**.xlsx` contém vários tipos de dados arranjados em matrizes $n \times m$ que incluem dados de abundância de espécies em diferentes unidades amostrais (UA's), dados da estrutura do habitat físico, e variáveis em escala de bacia hidrográfica, dados de contagem de indivíduos ajustados para Captura Por Unidade de Esforço (CPUE), etc (Figura 3.1).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	exto	ap-davis	ap-bimes	ap-faci	ch-bimac	cl-ocell	cl-orient	co-macro	co-heter	cr-menoz	cu-lepid	cy-gibe	ge-brasil	he-margi	ho-malab	hy-pusar	le-melan	la-pluv
2	S-A-ZAI	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	S-R-CCL	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	S-R-CT1	0	194	55	0	0	0	5	0	1	14	0	0	3	0	1	9	0
5	S-R-CT2	0	19	0	0	0	0	0	0	0	0	0	0	0	0	5	2	3
6	S-A-TA1	0	23	1	13	0	0	0	0	0	0	0	0	0	0	0	0	0
7	S-R-CT2	0	142	3	3	0	69	0	0	4	0	0	0	1	17	43	0	1
8	S-R-CP2	0	5	1	0	40	9	0	0	0	0	0	0	0	10	2	0	3
9	S-R-TC2	0	46	0	176	0	0	0	0	0	0	0	0	0	2	0	0	0
10	S-R-CT3	0	206	66	0	0	25	0	0	0	0	0	0	1	31	11	0	2
11	S-R-CP3	0	16	0	0	13	24	0	0	0	0	0	0	0	4	0	0	1
12	S-A-TA3	0	234	7	238	0	0	2	0	0	0	0	0	0	20	0	0	0
13	S-R-CT4	0	0	1	0	0	5	0	0	1	0	0	50	3	1	4	3	0
14	S-R-CP4	0	0	0	0	11	6	0	0	0	0	0	0	0	2	2	0	2
15	S-A-TA4	0	394	0	273	0	0	0	0	1	0	0	1	0	9	0	0	2
16	S-A-MU1	0	12	0	0	0	0	0	0	0	0	0	190	0	0	0	0	0
17	S-R-ET1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	S-A-GU1	0	2	2	0	0	0	0	0	0	0	0	7	0	0	0	0	0
19	S-R-PC2	0	441	0	0	2	0	0	0	0	0	0	8	0	0	0	2	0
20	S-A-MU2	0	99	0	0	0	0	0	0	0	0	0	67	0	1	0	0	0
21	S-A-GU2	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0
22	S-R-PC1	22	75	7	0	4	0	0	0	0	21	0	0	16	0	0	2	1
23	S-A-MU3	0	511	0	0	4	0	0	0	0	0	0	145	0	0	0	0	0
24	S-A-GU3	0	6	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0
25	S-R-PC4	0	7	17	0	0	0	0	0	0	0	0	81	5	0	1	0	1
26	S-A-MU4	0	235	0	0	0	0	0	0	0	0	0	509	0	0	0	0	0
27	S-A-GU4	0	13	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0

Figure 3.1: Parte da planilha de dados brutos do PPBio.

Por exemplo, essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados para os valores de Captura Por Unidade de Esforço (CPUE), nem foram relativizados ou transformados. Outros tipos de arquivos existem sobre esses dados (Tabela ??).

Várias das espécies nessa matriz tem grande importância ecológica, como é o caso de *Astyanax bimaculatus*¹ (Figura 3.2), que é muito comum em rios intermitentes e serve de alimento para predadores maiores (SILVA; DUARTE; MEDEIROS, 2018) como a espécie *Hoplias malabaricus*² (Figura 3.3) (SILVA et al., 2010).

Outras espécies como *Apareiodon hasemani*³ (Figura 3.4 tem importância trófica por estar na base da cadeia alimentar, enquanto espécies da família Loricariidae, como *Pseudancistrus genisetiger*⁴ (Figura 3.5), tem importância para taxonomia.

¹A etimologia do gênero *Astyanax* vem da mitologia Grega. Heitor personagem da “Ilíada”, tinha um filho chamado Astianax.

²Do Grego, *hoplon*, arma ou armadura, em referência aos dentes caniniformes muito desenvolvidos, e forte estrutura óssea na cabeça.

³A etimologia do nome *Apareiodon* vem do Grego, *a*, sem, *pareia*, lateral ou bochecha, *odus* dentição, em referência a ausência de dentes laterais no aparato bucal dessa espécie.”

⁴A etimologia do nome *Pseudancistrus* vem do Grego, *pseudes*, falso, e *agkistron*, gancho, em referência a falsos ganchos presentes na cabeça em algumas espécies do gênero.



Figure 3.2: **Astyanax bimaculatus**, a espécie mais comum da matriz de dados ppbio. Peru, by Eakins, R. Fonte: <<https://www.fishbase.se/summary/Astianax-bimaculatus.html>>



Figure 3.3: **Hoplias malabaricus**, espécie que cresce para se tornar um importante predador. Brazil, by Roselet, F.F.G. Fonte: <<https://www.fishbase.se/summary/Hoplias-malabaricus.html>>



Figure 3.4: **Apareiodon* sp.*, importante espécie bentopelágica das bacias dos rios Jaguaribe e Paraíba. Brazil, by Ramos, T.P.A. Fonte: <<https://www.fishbase.se/summary/Apareiodon-davisi.html>>



Figure 3.5: **Pseudancistrus genisetiger**, uma espécie endêmica das bacias hidrográficas do nordeste. By Medeiros, E.S.F. Fonte: Arquivo pessoal

As planilhas `ppbio**.xlsx` contém o delineamento amostral de um dos estudos do Projeto PPBio (Figura 3.6). Nas linhas são apresentadas as abreviações dos nomes das unidades amostrais (UA's) e nas colunas são apresentados os nomes abreviados das espécies - temos portanto uma matriz comunitária (Tabela ??). No corpo da planilha temos os valores para o tipo de dados amostrado. Quantitativo, semi-quantitativo ou qualitativo.

Qual desses tipos de dados você acha que é apresentado na planilha?

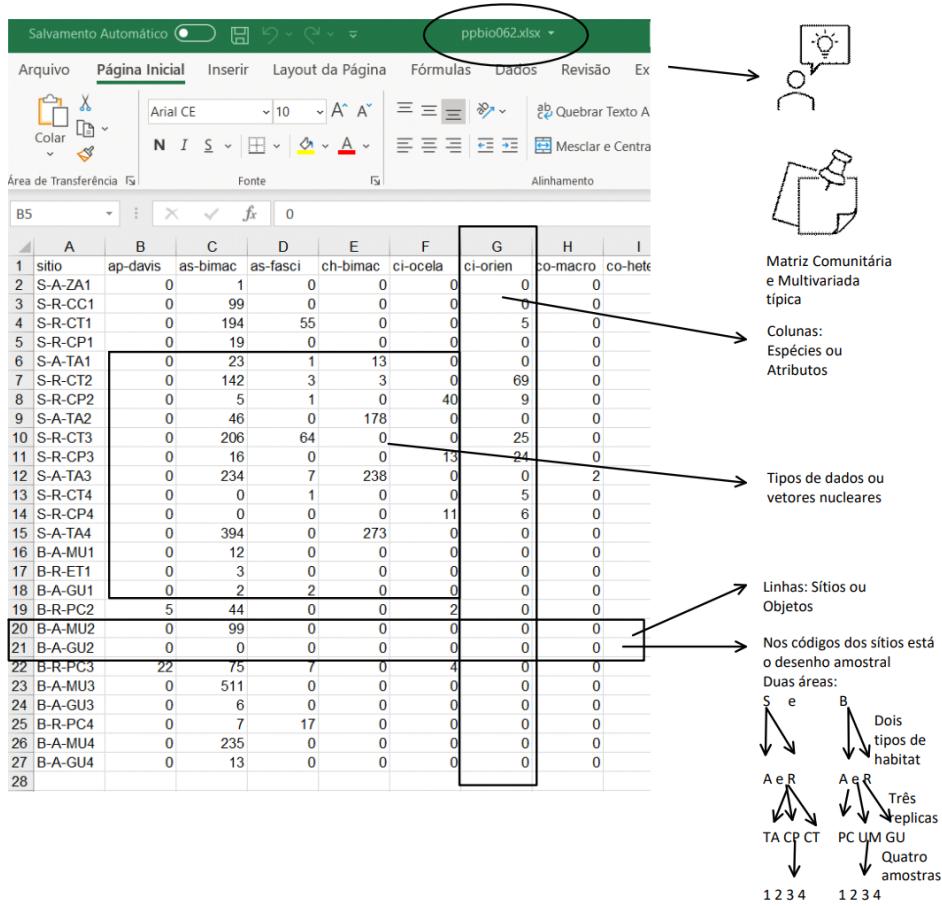


Figure 3.6: Associação entre a planilha de dados brutos do PPBio e o delineamento amostral do estudo.

3.2 Arquivos disponíveis

A seguir, apresento uma tabela com as informações essenciais sobre as matrizes de dados que serão utilizadas ao longo deste livro (Tabela ??). Nela, estão

descritos os diferentes tipos de matrizes, suas finalidades analíticas e o tipo de dado para cada uma delas. Essas informações servirão de referência para compreender a origem, a estrutura e o tratamento recomendado para os dados empregados nas análises subsequentes.

Você pode baixar essas matrizes clicando no link para o arquivo em **.xlsx** na coluna **Arquivo**. Ao vizualizar a matriz que deseja baixar, clique em **Arquivo > Baixar > Microsoft Excel (.xlsx)**. Atente para a pasta onde seu computador baixa o arquivo desejado, você vai precisar dessa informação depois.

Arquivo	Tipo de matriz	Descrição	Tipo de dados
ppbio06c-peixes.xlsx; comu- ppbio06c-peixes.ods	Matriz nitária	O arquivo ppbio06 traz os dados brutos que serão usados nas análises. A matriz de dados brutos contendo 26 localidades em estações do ano diferentes (objetos) x 35 espécies (atributos), antes de qualquer modificação.	Contagens de indivíduos com alta amplitude de variação, sugerido uso de matriz relativizada.
ppbio06p-amb	Matriz ambi- ental	O arquivo ppbio06h traz os dados brutos que serão usados nas análises. A matriz de dados brutos contendo 26 localidades em estações diferentes (objetos) x 35 variáveis ambientais (atributos) medidas em diferentes escalas espaciais, antes de qualquer modificação.	Unidades de medição diferentes (cm, m, °C, mg/L, etc.), com alta amplitude de variação, sugerido uso de matriz transformada e/ou reescalada.
ppbio06-grupos	Matriz de grupos	O arquivo ppbio06 traz os dados brutos que serão usados nessa análise. A matriz de dados brutos contendo 26 locais/ocasiões (objetos) x 35 espécies (atributos), antes de qualquer modificação.	Contagens de indivíduos com alta amplitude de variação, sugerido uso de matriz relativizada.

ppbio06cpuMatriz peixes.xlsx comu- nitária	O arquivo ppbio06cpue traz os valores após ajuste pela Captura Por Unidade de Esforço (CPUE).	Densidades de indivíduos com alta amplitude de variação, sugerido uso de matriz relativizada.
--	--	--

3.2.1 Codificação das variáveis

Os arquivos da base de dados do projeto são fornecidos em formato Excel (.xlsx). Por exemplo, o arquivo `ppbio06-* .xlsx`, traz os dados brutos que serão usados nas análises. A matriz de dados brutos contem mais de 20 localidades (m=linhas ou objetos) em estações do ano diferentes, e cerca de 35 espécies (n=colunas ou atributos), antes de qualquer modificação. Portanto é uma matriz bruta. Os valores são contagens de indivíduos, e apresentam uma alta amplitude de variação, portanto, o uso de uma matriz relativizada é sugerido (Tabela ??). Nos nomes dos objetos e dos atributos são codificados de acordo com a tabela mostrada na Figura 3.7.

Espécies de peixes	Sítios e período de amostragem														
ap-davis: Apareiodon hasemani as-bimac: Astyanax bimaculatus as-fasci: Astyanax fasciatus ch-bimac: Characidium bimaculatum ci-ocela: Cichla ocellaris ci-orient: Cichlasoma orientale co-macro: Colossoma macropomum co-hetero: Compurra heterura cr-menez: Crenicichla menezesi cu-lepid: Curimatella lepidura cy-gilbe: Cyphocharax Gilbert ge-brasi: Geophagus brasiliensis he-margi: Hemigrammus marginatus ho-malab: Hoplias malabaricus hy-pusar: Hypostomus pusarum le-melan: Leporinus melanopleura le-piau: Leporinus piau le-taeni: Leporinus taeniatus mo-costa: Moenkhausia costae mo-lepid: Moenkhausia sp1 or-nilot: Oreochromis niloticus pa-manag: Parachromis managuensis pimel-sp: Pimelodella enochi po-retic: Poecilia reticulata po-vivip: Poecilia vivipara pr-brevi: Prochilodus brevis ps-lsango: branquinha ps-rhomb: Psectrogaster rhomboidea ps-genise: Pseudancistrus papariae se-heter: Serrapinnus heterodon se-piaba: Serrapinnus piaba se-spiolo: Serrasalmus rhombeus st-noton: Steindachnerina notonota sy-marmo: Synbranchus marmoratus te-chalc: Tetragonopterus chalceus tr-signa: Triportheus signatus	<p>Nos códigos dos sítios está o desenho amostral. Duas áreas, ao norte o Seridó (S) e ao sul Buíque (B). Em cada área existem dois tipos de ambiente, açude (A) e rio (R). Ao norte estão três sítios, TA, CP e CT, e ao sul três sítios, PC, MU e GU. Em cada sítio há quatro amostragens por ano, trimestre 1, 2, 3 e 4.</p> <p>Resumo:</p> <table style="margin-left: 20px;"> <tr><td>S: Seridó</td></tr> <tr><td>B: Buíque</td></tr> <tr><td>A: Açude</td></tr> <tr><td>R: Rio</td></tr> </table> <table style="margin-left: 20px;"> <tr><td>TA: Açude Recanto</td></tr> <tr><td>CP: Riacho Cipó</td></tr> <tr><td>CT: Riacho Catureré</td></tr> <tr><td>PC: Riacho Poço da Cruz</td></tr> <tr><td>MU: Açude Mulungu</td></tr> <tr><td>GU: Açude Gurjão</td></tr> </table> <table style="margin-left: 20px;"> <tr><td>1: 1º. trimestre</td></tr> <tr><td>2: 2º. trimestre</td></tr> <tr><td>3: 3º. trimestre</td></tr> <tr><td>4: 4º. trimestre</td></tr> </table> <p>Exemplo:</p> <p>A amostra S-R-CP-1 representa, a região Seridó (S), um ambiente de rio (R), o sítio riacho Cipó (CP), no primeiro trimestre do ano (1).</p>	S: Seridó	B: Buíque	A: Açude	R: Rio	TA: Açude Recanto	CP: Riacho Cipó	CT: Riacho Catureré	PC: Riacho Poço da Cruz	MU: Açude Mulungu	GU: Açude Gurjão	1: 1º. trimestre	2: 2º. trimestre	3: 3º. trimestre	4: 4º. trimestre
S: Seridó															
B: Buíque															
A: Açude															
R: Rio															
TA: Açude Recanto															
CP: Riacho Cipó															
CT: Riacho Catureré															
PC: Riacho Poço da Cruz															
MU: Açude Mulungu															
GU: Açude Gurjão															
1: 1º. trimestre															
2: 2º. trimestre															
3: 3º. trimestre															
4: 4º. trimestre															

Figure 3.7: Codificação para as variáveis, espécies de peixes, sítios de amostragem e período de amostragem

3.2.2 Abreviações

No interesse de sistematizar o código R das várias matrizes que são comumente usadas em uma AMD, a tabela 3.2, a seguir, resume seus tipos e abreviações.

```
## Warning in table_info$align_vector[column] <-  
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para  
## para substituir não é um múltiplo do comprimento do substituto
```

Referências

Table 3.2: Nomenclatura das matrizes em AMD em relação aos atributos das colunas e suas abreviações nos códigos do R.

Nome	Atributos (colunas)	Abreviação no R
Matriz comunitaria	Os atributos são táxons ou OTU's (Unidades Taxonômicas Operacionais) (ex. espécies, gêneros, morfotipos)	m_com
Matriz ambiental	Os atributos são dados ambientais e variáveis físicas e químicas (ex. pH, condutividade, temperatura)	m_amb
Matriz de habitat	Os atributos são elementos da estrutura do habitat (ex. macróficas, algas, pedras, lama, etc)	m_hab
Matriz bruta	Os atributos ainda não receberam nenhum tipo de tratamento estatístico (valores brutos, como coletados)	m_bruta
Matriz transposta	Os atributos foram transpostos para as linhas	m_t
Matriz relativizada	Os atributos foram relativizados por um critério de tamanho ou de variação (ex. dividir os valores de cada coluna pela soma)	m_rel, m_relcol, m_rellin
Matriz transformada	Foi aplicado um operador matemático a todos os atributos (ex. raiz quadrada, log)	m_trns, m_log10, m_asrq
Matriz de distâncias	Matriz de m x m similaridades ou de distâncias (ex. Euclidiana, Manhattan, Bray-Curtis, etc)	m_dists, m_euclid, m_bray
Matriz de trabalho	Qualquer matriz que seja o foco da análise atual (ex. comunitária, relativizada, etc)	m_trab
Matriz particionada	Foram removidas linhas ou colunas (ex. linhas que são outliers e espécies zeradas)	m_part
Base de dados	Arquivo do Excel planilhado a partir de dados de campo ou de laboratório. Será manejada e particionada no R, para criar a Matriz bruta	ppbio06.xlsx, zoore- bio.xlsx, ben- tos06.xlsx

Chapter 4

Base de dados I - Peixes

Estrutura da Comunidade: Peixes no semiárido

Apresentação

Exemplo de conjunto de dados ecológicos do Programa de Pesquisa em Biodiversidade (PPBio) do Semiárido. Ano de 2006. Peixes¹

Autor:

- Prof. Elvio S. F. Medeiros
- Laboratório de Ecologia
- Universidade Estadual da Paraíba
- Campus V, João Pessoa, PB

Para entender a distribuição das espécies de peixes e seu uso de habitat, uma série de variáveis ambientais foram avaliadas como preditores da composição e riqueza da assembleia de peixes em sistemas aquáticos tropicais semiáridos. Nós pesquisamos a composição de espécies de assembleias de peixes em sistemas aquáticos semiáridos e estabelecemos seu grau de associação com a estrutura do habitat aquático. Os locais consistiam em trechos de riachos com fluxo de água superficial, poças temporárias isoladas e reservatórios artificiais (açudes). A amostragem de peixes foi realizada em quatro ocasiões durante as estações chuvosa (abril e junho de 2006) e seca (setembro e dezembro de 2006).

Texto modificado de MEDEIROS et al. (2024): Environmental variables as predictors of fish community composition in semiarid aquatic systems. Acta Limnologica Brasiliensis, 2024, vol. 36, e4. <https://doi.org/10.1590/S2179-975X3023>. ISSN 2179-975X on-line version. Tradução: Google Translator.

¹Projeto financiado pelo CNPq/UEPB/DCR Proc.350082/2006-5, UEPB/FAPESQ Proc.68.0006/2006.0 e Projeto de Pesquisa em Biodiversidade do Semiárido (PPBio SemiÁrido).

Elvio Sergio Figueiredo Medeiros¹, Marcio Joaquim da Silva², Telton Pedro Anselmo Ramos² e Robson Tamar Costa Ramos^{1,3}

¹ Grupo Ecologia de Rios do Semiárido, Universidade Estadual da Paraíba, Depto. de Biologia. Campus V. CEP 58070-450 João Pessoa - PB. Brazil. ² Universidade Federal do Rio Grande do Norte, Depto. de Biologia, Centro de Biociências, CEP 59078-900 – Natal - RN. ³ Laboratório de Sistemática e Morfologia de Peixes, Universidade Federal da Paraíba, Depto. de Sistemática e Ecologia, CCEN, CEP 58059-900, João Pessoa - PB.

- Autor correspondente: Dr. Elvio Medeiros <https://orcid.org/0000-0002-7472-8147>. Universidade Estadual da Paraíba, João Pessoa - Paraíba – Brasil - CEP 58.070-450. E-mail: elviomedeiros@servidor.uepb.edu.br

RESUMO

Para compreender a distribuição das espécies de peixes e o uso do habitat, uma série de variáveis ambientais foram avaliadas como preditores da composição de montagem de peixes e riqueza em sistemas aquáticos semiáridos tropicais. Pesquisamos a composição das espécies de conjuntos de peixes em sistemas aquáticos semiáridos e estabelecemos seu grau de associação com a estrutura do habitat aquático. Os locais consistiam em córregos com fluxo de água superficial, piscinas temporárias isoladas e reservatórios feitos pelo homem. A amostragem de peixes foi realizada em quatro ocasiões durante as estações úmidas (abril e junho de 2006) e secas (setembro e dezembro de 2006). A regressão múltipla de tepwise mostrou que as variáveis morfométricas, largura de alcance do córrego, comprimento do córrego e elevação explicaram 75,6% da variação da riqueza dos peixes. Cobertura de macrófitas e vegetação pendente somado ao poder preditivo da equação do modelo, onde o modelo final explicou 86,9% da variação da riqueza dos peixes. A Análise de Correspondência Canônica mostrou uma relação significativa entre os dados de composição do peixe e a morfologia do local (altitude, inclinação do banco e profundidade litorânea). Entre a qualidade da água, a composição do habitat e as variáveis substrato, temperatura, areia e cascalho apresentaram maior correlação com os eixos CCA. Esses resultados indicam que as comunidades de peixes assumem diferentes estruturas e composições em diferentes tipos de habitat, de acordo com a heterogeneidade ambiental nos sistemas aquáticos de terra seca.

Palavras-chave: fluxos intermitentes, reservatórios, conservação, composição de substratos.

4.1 Introdução

Padrões de distribuição de espécies de peixes dentro dos ecossistemas são frequentemente explicados em termos da estrutura do habitat disponível (Junqueira et al., 2016). O habitat é frequentemente relatado como estruturas subaquáticas físicas, como rochas, madeira submersa, macrófitas, algas, etc,

e como cobertura (ou proteção aérea) fornecida por características litorâneas, que incluem vegetação costeira ou troncos pendurados nas margens do córrego (Stewart-Koster et al., 2007). Indiscutivelmente, a qualidade e a quantidade do habitat afetam a estrutura e a composição das comunidades de peixes (Vono e Barbosa, 2001), por mudanças nas profundidades da água, velocidade e, consequentemente, tipo substrato, características litorâneas (como vegetação suspensa) e estruturas subaquáticas (Stewart-Koster et al., 2007, Medeiros e Arthington, 2011). Comumente, esses fatores são integrados e os habitats tendem a variar e segregar-se em manchas hierárquicas discretas (Frissell et al., 1986). Apesar disso, o habitat físico dos sistemas aquáticos de água doce em todo o mundo foi degradado ou modificado pelas atividades humanas (Hall et al., 2002). Tal interferência muitas vezes resulta da conversão de sistemas lóticos em lênticos, que mudam a dinâmica do habitat e seu funcionamento (Bunn e Arthington, 2002).

Uma série de estudos relacionam a estrutura da comunidade biótica (número de espécies e distribuição de abundância) e o habitat marginal físico. Tais estudos mostram que tipos de habitat mais complexos fornecem condições favoráveis para os peixes, como substrato de crescimento, locais de desova, bem como alimentos e proteção contra predação (Pusey e Arthington, 2003, Cucherousset et al., 2007, Jeffres et al., 2008) e, portanto, uma maior diversidade de micro-habitats associados às características litorâneas tem sido ligada a uma maior diversidade de espécies (Casatti et al., 2012). Como resultado, o fato de diferentes espécies de peixes apresentarem preferências particulares para diferentes tipos de habitat, cria padrões locais de composição de conjuntos de espécies (Junqueira et al., 2016). No entanto, o que leva a presença de espécies a tipos específicos de habitat muitas vezes não é claro.

Nos sistemas de riacho em regiões secas, espera-se que suas amplas variações espaciais e temporais levem a habitats físicos variáveis e, portanto, padrões específicos de estrutura de habitat (Hodges e Magoullick, 2011). Espera-se que a riqueza e a composição dos peixes sigam tais mudanças. Nesses sistemas, o habitat físico é espacial e temporalmente dinâmico devido à interação da morfologia do canal de córrego (por exemplo, largura, profundidade e substrato) e da variabilidade hidrológica (Medeiros et al., 2008, Farias et al., 2012). O estado do habitat físico a um determinado alcance será, portanto, influenciado por fatores que operam em diferentes escalas (Meninos e Thoms, 2006). No nível de captação, geomorfologia e clima afetam hidrologia, deposição de sedimentos, insumos de nutrientes e morfologia do canal (Davies et al., 2000, Mugodo et al., 2006). A nível local, características físicas e químicas da água, o uso da terra e o manejo da terra influenciarão o habitat de alcance do córrego (Hodges e Magoullick, 2011).

É importante ressaltar que a relação entre variáveis de habitat físico e riqueza e distribuição de espécies de peixes tem sido investigada em sistemas lúdicos e lóticos tropicais (por exemplo, Vono e Barbosa, 2001, Casatti et al., 2012), mas essa relação para as comunidades de peixes semiáridos permanece amplamente

desconhecida. Este trabalho mede a riqueza de peixes e a composição de espécies em diversos habitats aquáticos em córregos intermitentes do Semiárido Brasil, a fim de avaliar a composição das espécies de conjuntos de peixes e estabelecer seu grau de associação com a estrutura do habitat aquático. Temos a hipótese de que a fauna de peixes é irregular com as montagens representando características locais do habitat e que as variáveis ambientais serão elementos importantes explicando a distribuição de peixes.

4.1.1 Desenho Amostral

Este estudo foi realizado na região semiárida brasileira, em bacias hidrográficas que fluem através de uma floresta aberta arbustiva seca decidual (a “Caatinga”). A amplitude térmica é baixa na área de estudo, com médias variando de aproximadamente 25 a 30 °C, e a precipitação média anual varia entre 600 e 1100 mm. As altitudes variam entre 100 e 1000 m (Figura 4.1).

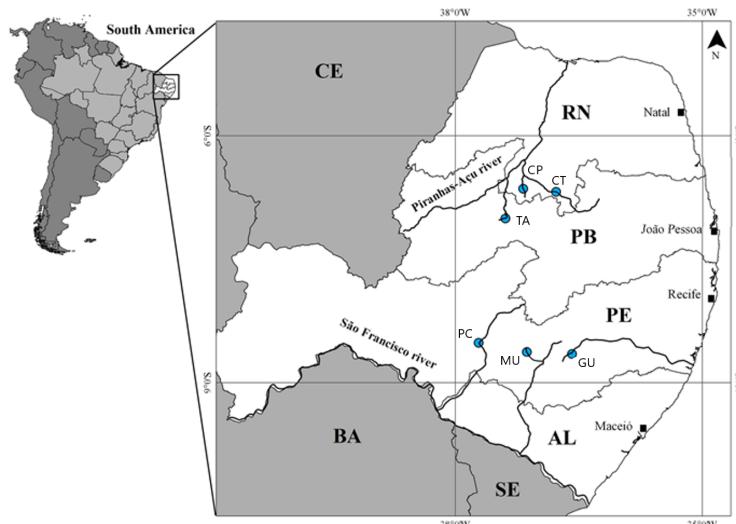


Figure 4.1: Área de estudo mostrando os estados do Rio Grande do Norte (RN), Paraíba (PB), Pernambuco (PE) e Alagoas (AL), principais sistemas fluviais e pontos de amostragem na região semiárida do Brasil. TA, açude Recanto; CP, riacho Cipó; CT, riacho Catureré; PC, riacho Poço da Cruz; MU, açude Mulungu e GU, açude Gurjão.

Duas regiões de importância biológica (**Seridó, S e Buíque, B**) foram escolhidas e nelas, **seis locais** ou sítios de coleta (TA, CP, CT, PC, MU, e GU) foram selecionados para representar ambientes Unidades Amostrais de ambientes temporários naturais e artificiais típicos (Figura 4.1 e Figura 4.6).

Os locais consistiam em trechos de **riachos (-R)** com fluxo de água superficial (durante a estação chuvosa) ou poças temporárias isoladas (durante a estação

seca) e **reservatórios ou açudes (-A)** artificiais criados a partir do representamento de riachos. A amostragem foi realizada durante o ano de 2006 em **quatro ocasiões** durante as estações chuvosa (**abril, 1 e junho, 2**) e seca (**setembro, 3 e dezembro, 4**) (Figura 4.1 e Figura 4.6).

Várias das espécies desse estudo tem grande importância ecológica, como é o caso de *Astyanax bimaculatus*² (Figura 4.2), que é muito comum em rios intermitentes e serve de alimento para predadores maiores como a espécie *Hoplias malabaricus*³ (Figura 4.3).



Figure 4.2: **Astyanax bimaculatus**, a espécie mais comum da matriz de dados ppbio. Peru, by Eakins, R. Fonte: <<https://www.fishbase.se/summary/Astianax-bimaculatus.html>>

Outras espécies como *Apareiodon hasemani*⁴ (Figura 4.4 tem importância trófica por estar na base da cadeia alimentar, enquanto espécies da família Loricariidae, como *Pseudancistrus genisetiger*⁵ (Figura 4.5), tem importância para taxonomia.

4.2 Codificação das variáveis

Os arquivos da base de dados do projeto são fornecidos em formato Excel (.xlsx). Por exemplo, o arquivo **ppbio06*-peixes.xlsx**, traz os dados brutos que serão

²A etimologia do gênero *Astyanax* vem da mitologia Grega. Heitor personagem da “Ilíada”, tinha um filho chamado Astíanax.

³Do Grego, *hoplon*, arma ou armadura, em referência aos dentes caniniformes muito desenvolvidos, e forte estrutura óssea na cabeça.

⁴A etimologia do nome *Apareiodon* vem do Grego, *a*, sem, *pareia*, lateral ou bochecha, e *odous* dentição, em referência a ausência de dentes laterais no aparato bucal dessa espécie.

⁵A etimologia do nome *Pseudancistrus* vem do Grego, *pseudes*, falso, e *agkistron*, gancho, em referência a falsos ganchos presentes na cabeça em algumas espécies do gênero.



Figure 4.3: **Hoplias malabaricus**, espécie que cresce para se tornar um importante predador. Brazil, by Roselet, F.F.G. Fonte: <<https://www.fishbase.se/summary/Hoplias-malabaricus.html>>



Figure 4.4: **Apareiodon* sp.*, importante espécie bentopelágica das bacias dos rios Jaguaribe e Paraíba. Brazil, by Ramos, T.P.A. Fonte: <<https://www.fishbase.se/summary/Apareiodon-davisi.html>>



Figure 4.5: **Pseudancistrus genisetiger**, uma espécie endêmica das bacias hidrográficas do nordeste. By Medeiros, E.S.F. Fonte: Arquivo pessoal

usados nas análises. A matriz de dados brutos contém mais de 20 localidades ($n=$ linhas ou objetos) em estações do ano diferentes, e cerca de 35 espécies ($m=$ colunas ou atributos), antes de qualquer modificação. Portanto é uma matriz bruta. Os valores são contagens de indivíduos, e apresentam uma alta amplitude de variação, portanto, o uso de uma matriz relativizada é sugerido. Nos nomes dos objetos e dos atributos são codificados de acordo com a tabela mostrada na Figura 4.6.

4.3 Amostragem dos dados

As coletas de peixes foram realizadas utilizando-se quatro tipos diferentes de equipamentos de amostragem, durante o dia, e de acordo com Medeiros et al (2010):

- uma rede de arrasto curta (4 m de comprimento, 1,5 m de malha de altura e 5 mm),
- uma rede de arrasto longa (20 m de comprimento, 2 m de altura e 12 mm de malha),
- um conjunto de redes de espera (30 m de comprimento e 1,5 m de altura divididas em três painéis de 10 m de 35, 45 e 55 mm de malha), e
- uma rede de tarrafa (2,4 m de altura e 12 mm de malha).

Espécies de peixes	Sítios e período de amostragem														
ap-davis: Apareiodon hasemani as-bimac: Astyanax bimaculatus as-fasci: Astyanax fasciatus ch-bimac: Characidium bimaculatum ci-ocela: Cichla ocellaris ci-orien: Cichlasoma orientale co-macro: Colosoma macropomum co-hetero: Compurua heterura cr-menez: Crenicichla menezesi cu-lepid: Curimatella lepidura. cy-gilbe: Cyphocharax Gilbert ge-brasi: Geophagus brasiliensis he-margi: Hemigrammus marginatus ho-malab: Hoplias malabaricus hy-pusar: Hypostomus pusarum le-melan: Leporinus melanopleura le-piau: Leporinus piau le-taeni: Leporinus taeniatus mo-costa: Moenkhausia costae mo-lepidi: Moenkhausia sp1 or-nilot: Oreochromis niloticus pa-manag: Parachromis managuensis pimel-sp: Pimelodella enochi po-retic: Poecilia reticulata po-vivip: Poecilia vivipara pr-brevi: Prochilodus brevis ps-losango: branquinha ps-rhomb: Psectrogaster rhombooides ps-genise: Pseudancistrus papariae se-heter: Serrapinnus heterodon se-piaba: Serrapinnus piaba se-spilo: Serrasalmus rhombeus st-noton: Steindachnerina notonota sy-marmo: Synbranchus marmoratus te-chalc: Tetragonopterus chalceus tr-signa: Triplotheus signatus	<p>Nos códigos dos sítios está o desenho amostral. Duas áreas, ao norte o Seridó (S) e ao sul Buíque (B). Em cada área existem dois tipos de ambiente, açude (A) e rio (R). Ao norte estão três sítios, TA, CP e CT, e ao sul três sítios, PC, MU e GU. Em cada sítio há quatro amostragens por ano, trimestre 1, 2, 3 e 4.</p> <p>Resumo:</p> <table style="margin-left: 40px;"> <tr><td>S: Seridó</td></tr> <tr><td>B: Buíque</td></tr> <tr><td>A: Açude</td></tr> <tr><td>R: Rio</td></tr> </table> <table style="margin-left: 40px;"> <tr><td>TA: Açude Recanto</td></tr> <tr><td>CP: Riacho Cipó</td></tr> <tr><td>CT: Riacho Catureré</td></tr> <tr><td>PC: Riacho Poço da Cruz</td></tr> <tr><td>MU: Açude Mulungu</td></tr> <tr><td>GU: Açude Gurjão</td></tr> </table> <table style="margin-left: 40px;"> <tr><td>1: 1o. trimestre</td></tr> <tr><td>2: 2o. trimestre</td></tr> <tr><td>3: 3o. trimestre</td></tr> <tr><td>4: 4o. Trimestre</td></tr> </table> <p>Exemplo:</p> <p>A amostra S-R-CP-1 representa, a região Seridó (S), um ambiente de rio (R), o sítio riacho Cipó (CP), no primeiro trimestre do ano (1).</p>	S: Seridó	B: Buíque	A: Açude	R: Rio	TA: Açude Recanto	CP: Riacho Cipó	CT: Riacho Catureré	PC: Riacho Poço da Cruz	MU: Açude Mulungu	GU: Açude Gurjão	1: 1o. trimestre	2: 2o. trimestre	3: 3o. trimestre	4: 4o. Trimestre
S: Seridó															
B: Buíque															
A: Açude															
R: Rio															
TA: Açude Recanto															
CP: Riacho Cipó															
CT: Riacho Catureré															
PC: Riacho Poço da Cruz															
MU: Açude Mulungu															
GU: Açude Gurjão															
1: 1o. trimestre															
2: 2o. trimestre															
3: 3o. trimestre															
4: 4o. Trimestre															

Figure 4.6: Codificação para as variáveis, espécies de peixes, sítios de amostragem e período de amostragem.

Environmental variables as predictors of fish community...

Table 2. Abundance, percentage and frequency of occurrence (F.O.) of species collected in the study sites in the semi-arid region of Brazil. Serridó stream (SE), Cipó stream (CI) and Recanto reservoir (RE) (Piranhas-Açu catchment), Escamalépe stream (EP; Moçotó River), Mulungu reservoir (MU; Ipanema River) and Salobro reservoir (SA; Una River).

	SE	CI	RE	EP	MU	SA	%	F.O.
Characiformes								
Curimatidae								
<i>Curimatella lepidura</i> (Eigenmann & Eigenmann, 1889)	0	0	0	21	0	0	0.2	4.3
<i>Psectrogaster rhomboides</i> Eigenmann & Eigenmann, 1889	1	0	0	0	0	0	0.01	4.3
<i>Steindachnerina notonota</i> (Miranda-Ribeiro, 1937)	205	0	0	0	0	0	2.3	17.4
<i>Cyphocharax gibbert</i> (Quoy & Gaimard, 1824)	50	0	0	81	0	0	1.5	8.7
Parodontidae								
<i>Apareiodon hasemani</i> Eigenmann, 1916	0	0	0	27	0	0	0.3	8.7
Prochilodontidae								
<i>Prochilodus brevis</i> Steindachner, 1874	243	5	6	15	1	0	3.1	47.8
Anostomidae								
<i>Leporinus piav Fowler, 1941</i>	6	6	2	1	0	0	0.2	34.8
<i>Leporinus melanopleura</i> Günther, 1864	0	0	0	2	0	0	0.02	4.3
<i>Leporinus taeniatus</i> Lütken, 1875	0	0	0	1	0	0	0.01	4.3
Crenuchidae								
<i>Characidium bimaculatum</i> Fowler, 1941	3	0	702	0	0	0	8.1	21.7
Characidae								
<i>Astyanax bimaculatus</i> (Linnaeus, 1758)	542	40	697	126	857	21	26.1	87.0
<i>Astyanax fasciatus</i> (Cuvier, 1819)	123	1	8	24	0	2	1.8	43.5
<i>Colossoma macropomum</i> (Cuvier 1816) *	0	0	2	0	0	0	0.02	4.3
<i>Compsura heterura</i> Eigenmann, 1915	1	0	0	0	0	0	0.01	4.3
<i>Hemigrammus marginatus</i> Ellis, 1911	2	0	0	0	0	0	0.02	8.7
<i>Moenkhausia costae</i> (Steindachner, 1907)	0	0	0	1	0	0	0.01	4.3
<i>Moenkhausia</i> sp.	40	0	0	0	0	0	0.5	8.7
<i>Serrapinnus heterodon</i> (Eigenmann, 1915)	141	17	4	134	0	0	3.4	43.5
<i>Serrapinnus piaba</i> (Lütken, 1875)	68	0	0	0	0	0	0.8	4.3
<i>Serrasalmus rhombeus</i> (Linnaeus, 1766)	1	0	0	0	0	0	0.01	4.3
<i>Tetragonopterus chalceus</i> Spix & Agassiz, 1829	0	0	0	134	0	0	1.5	8.7
<i>Triportheus signatus</i> (Garman, 1890)	181	0	0	27	0	0	2.4	26.1
Erythrinidae								
<i>Hoplias malabaricus</i> (Bloch, 1794)	53	21	31	3	1	0	1.2	60.9
Siluriformes								
Loricariidae								
<i>Hypostomus pusarum</i> (Starks, 1913)	66	4	0	1	0	0	0.8	30.4
<i>Pseudancistrus papirae</i> Fowler 1941	1	0	0	0	0	0	0.01	4.3
Heptapteridae								
<i>Pimelodella enochi</i> Fowler, 1941	6	0	0	0	0	0	0.1	4.3
Cyprinodontiformes								
Poeciliidae								
<i>Poecilia reticulata</i> Peters, 1860 *	25	0	0	0	322	0	4.0	21.7
<i>Poecilia vivipara</i> Bloch & Schneider, 1801	622	137	0	0	219	0	11.2	47.8
Synbranchiformes								
Synbranchidae								
<i>Synbranchus marmoratus</i> Bloch, 1785	0	1	0	0	0	0	0.01	4.3
Perciformes								
Cichlidae								
<i>Cichla ocellaris</i> Bloch & Schneider, 1801 *	0	64	0	6	0	0	0.8	21.7
<i>Cichlasoma orientale</i> Kullander, 1983	104	39	0	0	0	0	1.6	30.4
<i>Crenicichla meienezei</i> Ploeg, 1991	27	0	1	0	0	0	0.3	21.7
<i>Geophagus brasiliensis</i> (Quoy & Gaimard, 1824)	7	0	1	29	911	72	11.7	65.2
<i>Oreochromis niloticus</i> (Linnaeus, 1758) *	324	0	1	79	19	415	9.6	69.6
<i>Parachromis managuensis</i> (Günther, 1867) *	0	0	0	0	0	553	6.3	17.4
Total	2842	335	1455	712	2330	1063		

* = Non-native species.

Acta Limnologica Brasiliensis, 2024, vol. 36, e4

Figure 4.7: Tabela 1. Abundância, percentual e frequência de ocorrência (F.O.) de espécies coletadas nos locais de estudo (codificados conforme a Figura reffig:201comcodes) no semiárido do Brasil. = Espécies não nativas.

O esforço de captura foi semelhante em ocasiões e locais de amostragem, e medido em horas (para o método passivo), representando o número de horas das redes de espera na água, ou réplicas (para métodos ativos), representando o número de arrastos (para redes de arrasto) ou o número de lançamentos (para a tarrafa) (Medeiros et al., 2010). Os arrastos foram semelhantes em todas as ocasiões e locais de amostragem, sendo aproximadamente 10 m de comprimento para a grande rede de arrasto e 3 a 5 m de comprimento para a rede de arrasto curta.

Os peixes capturados foram eutanasiados em solução de eugenol (Resolução Normativa CONCEA nº 37, de 15.02.2018) fixados em formalia a 10% neutralizada com tetraborato de sódio e posteriormente transferidos para solução de 75% de etanol. Os espécimes foram tratados de acordo com as normas brasileiras de curadoria científica (Malabarba e Reis, 1987) e os peixes foram coletados sob a licença nº 032DIFAP/IBAMA do Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis (032DIFAP/IBAMA) de 23 de março de 2006. A triagem e identificação dos espécimes foram realizadas no Laboratório de Morfologia e Taxonomia de Peixes da Universidade Federal da Paraíba. Os espécimes-testemunho foram depositados após identificação na Coleta Ictiológica da mesma instituição.

Além da estrutura da comunidade, a estrutura física do habitat foi medida como,

- (1) variáveis físicas e químicas,
- (2) a morfologia do fluxo,
- (3) composição do substrato, e
- (4) estrutura de habitat.

As variáveis físicas e químicas foram medidas utilizando-se equipamentos portáteis para pH (TECNOPON MPA-210), condutividade (S/cm) (TECNOPON MCS-150), oxigênio dissolvido (mg/L) e temperatura (°C) (Lutron DO-5510). A transparência (cm) foi medida utilizando-se um disco secchi, e a velocidade da água (m/s) foi estimada usando o método flutuante (Maitland, 1990). O alcance do córrego ou morfologia do local foi avaliado pela largura média (cm) e profundidade (cm) retiradas de três transectos colocados ao longo do alcance do córrego, piscina ou reservatório. A composição do substrato e a estrutura do habitat foram estimadas em 9 a 12 pontos de pesquisa de 1 m² medidos nas margens (ver Medeiros et al., 2008). Em cada ponto de pesquisa, foram estimadas as proporções (%) da composição de sedimentos (classificados como lama, areia, cascalho e seixos) e estruturas litorâneas e subaquáticas (por exemplo, macrófitas, capim, vegetação submersa, vegetação suspensa, folhiço, algas e galhos) (Pusey et al., 2004; Medeiros et al., 2008).

4.4 Análises estatísticas

Foram realizadas análises estatísticas univariadas sobre riqueza e abundância. A abundância foi padronizada por Unidade de Esforço de Captura (CPUE), onde o número de peixes capturados foi dividido pelo número de horas ou réplicas de cada técnica amostral em cada ocasião amostral e local (a partir de agora denominado abundância CPUE) (Medeiros et al., 2010). A riqueza dos peixes foi corrigida utilizando-se análises rarefação para o número médio de indivíduos de todas as ocasiões de amostragem utilizando PRIMER-e 5.0 (Clarke e Gorley, 2001). A abundância CPUE e a riqueza rarefacionada foram comparadas entre os locais de estudo usando a ANOVA one-way seguida de comparações múltiplas pós-hoc usando o teste HSD de Tukey ($=0,05$) (Zar, 1999). A abundância CPUE foi transformada pela raiz quadrada, e as variáveis ambientais foram $\log_{10}(x+1)$ transformadas para aumentar a normalidade e a homogeneidade das variâncias (Sokal e Rohlf, 1969, Maltchik et al., 2010). A correlação entre a riqueza rarefacionada e a abundância CPUE (variáveis dependentes) com a estrutura de habitat (variáveis independentes) foi avaliada utilizando-se a regressão múltipla (RM) com “forward selection”, onde as variáveis são inseridas no modelo com base na significância (probabilidade) do valor F (Sheridan e Lyndall, 2001, Maltchik et al., 2010).

Os padrões de variação na composição da comunidade de peixes entre os locais foram avaliados utilizando-se de Escalonamento Multidimensional Não-Métrico (NMDS), com base na distância relativizada de Bray-Curtis da matriz de dados transformada pelo arco-seno da raiz quadrada. O Procedimento de Permutação Multiresposta (MRPP) (Biondini et al., 1985, McCune e Grace, 2002) foi utilizado para testar a significância das diferenças na composição dos peixes entre locais/ocasiões de amostragem. O valor de “A” é apresentado como medida do grau de homogeneidade entre os grupos em comparação com a expectativa aleatória. Utilizou-se a Análise de Espécies Indicadoras (ISA) para determinar quais espécies foram indicadores significativos dos locais. O Valor Indicativo (IV) para cada espécie foi calculado utilizando-se o método de Dufrene e Legendre (1997). Este valor é testado para significância usando o teste de Monte Carlo (999 permutações). Para estabelecer correlações multivariadas entre a composição dos peixes e as variáveis ambientais, foi realizada uma Análise de Correspondência Canônica (CCA) (McCune e Grace, 2002). A matriz de dados foi centrada e normalizada, e as correlações testadas pelo teste de Monte Carlo com 999 permutações. As variáveis ambientais foram transformadas pelo $\log_{10}(x+1)$ (Maltchik et al., 2010). As análises estatísticas ($=0,05$) foram realizadas no PC-ORD 7.0 (McCune e Mefford, 1999).

4.5 Discussão

A riqueza das espécies e a composição comunitária dos peixes observados no presente estudo estão de acordo com outros estudos realizados no semiárido do Brasil (Medeiros e Maltchik, 2001a, Medeiros et al., 2010) com a família Characi-

dae sendo a mais representativa em termos de riqueza e abundância. Somado ao fato de que a ordem Characiformes inclui um grande número de espécies de peixes do semiárido brasileiro (Rosa et al., 2003), a maioria das espécies de Characidae observadas no presente estudo são de pequeno porte com taxas relativamente altas de fecundidade e crescimento (Medeiros e Maltchik, 2000, 2001b) e generalistas em hábitos alimentares e no uso de habitat (Silva et al., 2010, Silva, 2012). Em comparação com outros sistemas de rios secos, os locais de estudo mostraram águas bem oxigenadas, transparentes e relativamente quentes ao longo do período de estudo (ver, por exemplo, Medeiros e Arthington, 2011). A morfologia do alcance do rio variou principalmente de acordo com a natureza do local de estudo, fluxo, piscina isolada ou reservatório, e o habitat aquático era diversificado com uma gama de elementos de habitat disponíveis para colonização pela biota aquática. Os córregos e piscinas de fluxo apresentaram uma maior variedade de elementos de habitat marginal e composição de substratos quando comparados aos reservatórios. Estudos no Semiárido Brasil têm demonstrado que o habitat aquático tende a ser mais rico em córregos, quando comparado aos reservatórios e defende que essa riqueza em estruturas subaquáticas e a natureza mais variável dos córregos intermitentes melhorem a diversidade de peixes (Medeiros et al., 2006, Medeiros et al., 2008).

A regressão múltipla mostrou que o rio atinge morfologia e a estrutura do habitat teve a maioria das variáveis explicando a riqueza dos peixes. O fato de que a morfologia do local é um fator importante que explica a riqueza dos peixes (principalmente largura, comprimento e elevação) indica que esses fatores espaciais estão associados a diferentes faixas de variáveis ambientais que suportam um número de espécies de peixes. Isso é apoiado pelas análises multivariadas que mostraram forte segregação espacial da comunidade de peixes em padrões de montagem espacial. Foi demonstrado que diferentes conjuntos de espécies de peixes exibem preferências por tipos específicos de habitat (Martin-Smith, 1998). No presente estudo, a ausência de fluxo de água na maioria dos locais e a falta de conectividade longitudinal levaram a condições ambientais específicas, segregando assim a comunidade de peixes aos padrões locais de composição das espécies. Portanto, na ausência de fluxo de água, espera-se que a comunidade de peixes assuma diferentes composições entre os tipos de habitat (ou seja, fluxo de vazão de córrego, piscina e reservatório) de acordo com a heterogeneidade ambiental. Isso é ainda mais apoiado pela riqueza significativamente diferente entre os locais, principalmente no que diz respeito aos córregos Seridó (SE) e Escama-Peixe (EP), que apresentaram maior riqueza média. Os resultados de abundância são menos conclusivos, uma vez que a ANOVA não mostrou diferença significativa entre os locais e a regressão múltipla relatou apenas oxigênio dissolvido e vegetação pendente como preditores importantes da abundância de peixes. As abundâncias naturais de peixes nos sistemas aquáticos semiáridos brasileiros variam muito, tanto espacialmente quanto temporalmente (Medeiros et al., 2010). Isso provavelmente está associado ao movimento de indivíduos dentro de pequenos trechos do rio e entre áreas mais rasas e mais profundas em piscinas ou reservatórios maiores durante inundações (Medeiros e Arthington,

2008), bem como com padrões de recrutamento e desova de peixes (Medeiros e Maltchik, 2000).

A importância da vegetação ribeirinha e sua influência no habitat aquático é bem reconhecida (Gregory et al., 1991), pois contribui com sombra e estruturas subaquáticas para refúgio de peixes, bem como locais de desova (Pusey e Arthington, 2003). No presente estudo, variáveis associadas à presença de vegetação ribeirinha foram importantes preditores da riqueza dos peixes (vegetação pendente) e associadas à composição da montagem (detritos lenhosos e temperatura da água). Essas variáveis provavelmente estão contribuindo com condições ambientais adequadas aos peixes, bem como criando embalagens locais de condições ambientais que contribuem com a segregação espacial observada na comunidade de peixes. A presença de macrófitos aquáticos em locais de reservatórios e córregos e sua importância como preditores para a riqueza e composição de espécies também é uma indicação do papel desempenhado pelas estruturas de habitat subaquático na criação de habitats locais espacialmente segregados disponíveis para uso de peixes (Xie et al., 2001). O habitat pode ser visto como um quadro onde variáveis ambientais afetam comunidades biológicas (Southwood, 1977). No entanto, em sistemas hidrologicamente variáveis, a variabilidade do fluxo tem sido sugerida como um grande eixo influenciando no modelo de habitat (Minshall, 1988, Poff e Ward, 1989). Os sistemas aquáticos de terra seca no Brasil são altamente variáveis no que diz respeito à presença e magnitude do fluxo de água (Maltchik e Florin, 2002) e, embora os resultados indiquem que a estrutura do habitat e da morfologia do local são importantes preditores da fauna de peixes, o modelo de habitat é multidimensional (Southwood, 1977) e as diversas variáveis ambientais medidas interagem entre si. Nos sistemas semiáridos do Brasil, a estrutura do habitat está sendo conduzida por muitos componentes em diferentes escalas (Medeiros et al., 2008), onde o papel das características de captação e morfologia local aumentaria na predominância em suas respectivas escalas. Isso destaca a importância da ligação entre a estrutura do habitat e a diversidade biótica no nível do habitat local.

Alterações nos padrões naturais de vazão de água com base na construção de reservatórios e açudes, decorrentes das políticas de gestão atuais dos sistemas semiáridos aquáticos no Brasil, modificam as características hidrológicas dos córregos intermitentes altamente variáveis (Leal et al., 2005, Maltchik e Medeiros, 2006). Com as mudanças na dinâmica natural do fluxo de água vêm também mudanças na estrutura de montagem de macrofílicas, dinâmica de nutrientes e conectividade longitudinal, todas associadas à conversão de sistemas loticos para lenticos (Bunn e Arthington, 2002). Os dados do presente estudo mostram que essas modificações afetarão as comunidades de peixes (riqueza e composição), uma vez que variáveis altamente associadas ao fluxo de água, como largura de alcance do córrego, cobertura de macrofíto, vegetação suspensa e concentração de oxigênio dissolvida foram importantes preditores da montagem de peixes. Portanto, a modificação dos padrões naturais de vazão da água e promoção de condições lenéticas tem o potencial de interferir na fauna de peixes, favorecendo espécies mais oportunistas mais adaptadas a condições de fluxo. Isso é corrobor-

orado pelo fato de que os locais dos reservatórios apresentaram menos espécies indicadoras que foram introduzidas principalmente, como *Poecilia reticulata* e *Parachromis managuensis*, ou tipicamente lenticas como *Geophagus brasiliensis*.

O presente estudo mostra que as comunidades de peixes assumem diferentes estruturas e composições em diferentes tipos de habitat, de acordo com a heterogeneidade ambiental associada à natureza do habitat (reservatórios, córregos e piscinas). A riqueza e a composição dos peixes foram influenciadas principalmente por dois conjuntos de variáveis ambientais, sendo morfologia do local e características do habitat. O primeiro conjunto de variáveis relaciona-se com os níveis hierárquicos espaciais de captação, uma vez que o comprimento e a elevação do rio estão tipicamente associados à hierarquia do rio. O segundo conjunto de variáveis é tipicamente local, com estruturas subaquáticas sendo resultado de processos locais como o fluxo de água de alcance do rio e o uso da terra/vegetação ribeirinha.

Apêndices

4.5.1 Tabelas e Figuras

Tabela 1. Abundância, percentual e frequência de ocorrência (F.O.) de espécies coletadas nos locais de estudo (codificados conforme a Figura 4.6) no semiárido do Brasil. * = Espécies não nativas.”

Tabela 2. Variáveis ambientais (average \pm SD) coletadas nos locais de estudo no semiárido do Brasil.

Tabela 3. São apresentadas as variáveis ambientais e a composição dos peixes no presente estudo (apenas variáveis correlacionadas com um determinado eixo são mostradas).

Figura 1. Área de estudo mostrando os estados do Rio Grande do Norte (RN), Paraíba (PB), Pernambuco (PE) e Alagoas (AL), principais sistemas fluviais e locais de amostragem no semiárido do Brasil. TA, açude Recanto; CP, riacho Cipó; CT, riacho Catureré; PC, riacho Poço da Cruz; MU, açude Mulungu e GU, açude Gurjão.

Figura 2. Os resultados da NMDS para a composição dos peixes em todo o estudo chegam ao semiárido do Brasil. Vetores (caixa de entrada) mostram taxa correlacionada ($r^2 > 0,2$) com amostras em espaço de ordenação. As letras após os códigos do site indicam ocasião de amostragem (A=Abri, J=Junho, S=Setembro e D=Dezembro).

Figura 3. Biplot de CCA mostrando a composição de conjuntos de peixes nos locais e ocasiões de amostragem () e as variáveis ambientais explicativas definidas pela CCA.

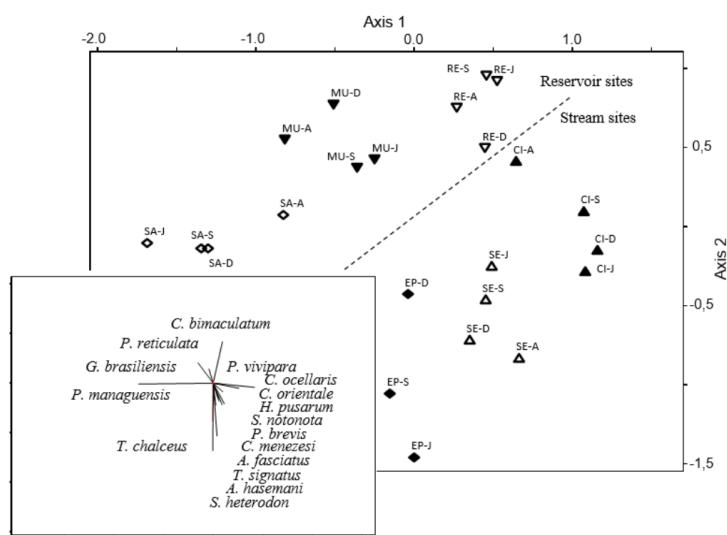
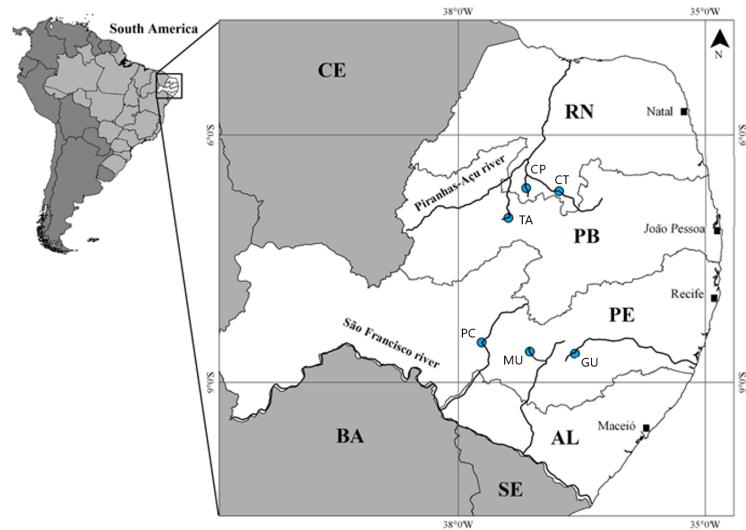
Environmental variables as predictors of fish community...
Table 2. Abundance, percentage and frequency of occurrence (F.O.) of species collected in the study sites in the semi-arid region of Brazil. Serrido stream (SE), Cipó stream (CI) and Recanto reservoir (RE) (Piranhas-Açu catchment), Escampe Peixe stream (EP; Moxotó River), Mulungu reservoir (MU, Ipanema River) and Salobro reservoir (SA, Una River).

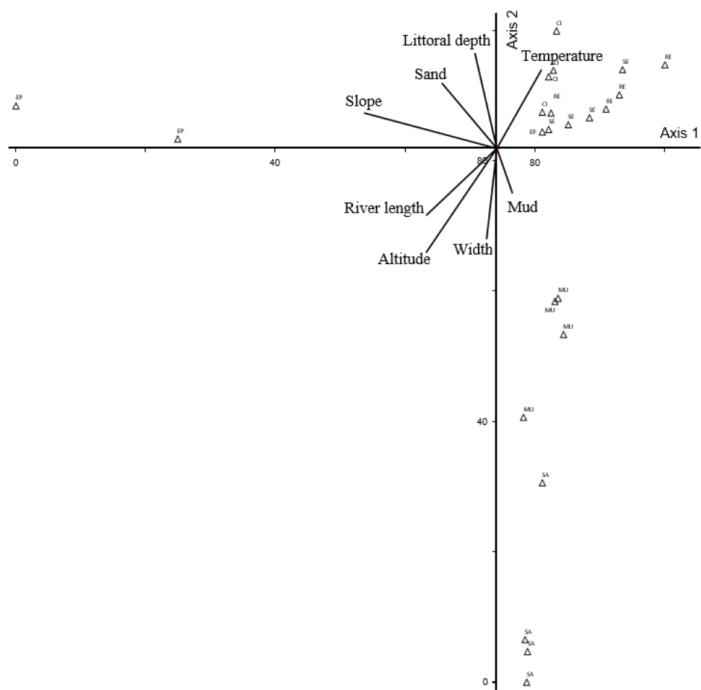
	SE	CI	RE	EP	MU	SA	%	F.O.
Characiformes								
Curimatidae								
<i>Curimatella lepidura</i> (Eigenmann & Eigenmann, 1889)	0	0	0	21	0	0	4.1	4.3
<i>Psectrogaster rhomboides</i> Eigenmann & Eigenmann, 1889	1	0	0	0	0	0	0.01	4.3
<i>Steindachnerina notonota</i> (Miranda-Ribeiro, 1937)	205	0	0	0	0	0	2.3	17.4
<i>Cyphocharax gibbert</i> (Quoy & Gaimard, 1824)	50	0	0	81	0	0	1.5	8.7
Parodontidae								
<i>Apareiodon hasemani</i> Eigenmann, 1916	0	0	0	27	0	0	0.3	8.7
Prochilodontidae								
<i>Prochilodus brevis</i> Steindachner, 1874	243	5	6	15	1	0	3.1	47.8
Anostomidae								
<i>Leporinus piav</i> Fowler, 1941	6	6	2	1	0	0	0.2	34.8
<i>Leporinus melanopleura</i> Günther, 1864	0	0	0	2	0	0	0.02	4.3
<i>Leporinus taeniatus</i> Lütken, 1875	0	0	0	1	0	0	0.01	4.3
Crenuchidae								
<i>Characidium bimaculatum</i> Fowler, 1941	3	0	702	0	0	0	8.1	21.7
Characidae								
<i>Astyanax bimaculatus</i> (Linnaeus, 1758)	542	40	697	126	857	21	26.1	87.0
<i>Astyanax fasciatus</i> (Cuvier, 1819)	123	1	8	24	0	2	1.8	43.5
<i>Colossoma macropomum</i> (Cuvier 1816) *	0	0	2	0	0	0	0.02	4.3
<i>Compura heterura</i> Eigenmann, 1915	1	0	0	0	0	0	0.01	4.3
<i>Hemigrammus marginatus</i> Ellis, 1911	2	0	0	0	0	0	0.02	8.7
<i>Moenkhausia costae</i> (Steindachner, 1907)	0	0	0	1	0	0	0.01	4.3
<i>Moenkhausia</i> sp.	40	0	0	0	0	0	0.5	8.7
<i>Serrapinnus heterodon</i> (Eigenmann, 1915)	141	17	4	134	0	0	3.4	43.5
<i>Serrapinnus piaba</i> (Lütken, 1875)	68	0	0	0	0	0	0.8	4.3
<i>Serrasalmus rhombeus</i> (Linnaeus, 1766)	1	0	0	0	0	0	0.01	4.3
<i>Tetragonopterus chalceus</i> Spix & Agassiz, 1829	0	0	0	134	0	0	1.5	8.7
<i>Triportheus signatus</i> (Garman, 1890)	181	0	0	27	0	0	2.4	26.1
Erythrinidae								
<i>Hoplias malabaricus</i> (Bloch, 1794)	53	21	31	3	1	0	1.2	60.9
Siluriformes								
Loricariidae								
<i>Hypostomus pusarum</i> (Starks, 1913)	66	4	0	1	0	0	0.8	30.4
<i>Pseudancistrus papirae</i> Fowler 1941	1	0	0	0	0	0	0.01	4.3
Heptapteridae								
<i>Pimelodella enochi</i> Fowler, 1941	6	0	0	0	0	0	0.1	4.3
Cyprinodontiformes								
Poeciliidae								
<i>Poecilia reticulata</i> Peters, 1860 *	25	0	0	0	322	0	4.0	21.7
<i>Poecilia vivipara</i> Bloch & Schneider, 1801	622	137	0	0	219	0	11.2	47.8
Synbranchiformes								
Synbranchidae								
<i>Synbranchus marmoratus</i> Bloch, 1785	0	1	0	0	0	0	0.01	4.3
Perciformes								
Cichlidae								
<i>Cichla ocellaris</i> Bloch & Schneider, 1801 *	0	64	0	6	0	0	0.8	21.7
<i>Cichlasoma orientale</i> Kullander, 1983	104	39	0	0	0	0	1.6	30.4
<i>Crenicichla menezesi</i> Ploeg, 1991	27	0	1	0	0	0	0.3	21.7
<i>Geophagus brasiliensis</i> (Quoy & Gaimard, 1824)	7	0	1	29	911	72	11.7	65.2
<i>Oreochromis niloticus</i> (Linnaeus, 1758) *	324	0	1	79	19	415	9.6	69.6
<i>Parachromis managuensis</i> (Günther, 1867) *	0	0	0	0	0	553	6.3	17.4
Total	2842	335	1455	712	2330	1063		

* = Non-native species.

	CI	SE	RE	SA	MU	EP
Morfologia						
Comprimento do rio (km)	83.1	163.2	110.2	212.7	214.0	196.8
Elevação (m)	169.0	226.0	270.0	713.0	725.0	402.0
Velocidade da água (m/s)	0,04 ($\pm 0,1$)	0,1 ($\pm 0,1$)	0,03 ($\pm 0,1$)	0.0	0.0	0.0
Inclinação máxima da margem	60.0	30.0	60.0	30.0	30.0	90.0
Profundidade máxima (cm)	67,8 ($\pm 8,2$)	98,8 ($\pm 16,6$)	117,0 ($\pm 27,9$)	65,0 ($\pm 4,2$)	124,0 ($\pm 18,8$)	95,0 ($\pm 15,0$)
Profundidade da margem (cm)	33,3 ($\pm 9,3$)	53,3 ($\pm 24,9$)	36,3 ($\pm 13,8$)	6,7 ($\pm 1,5$)	30,4 ($\pm 8,5$)	50,7 ($\pm 1,9$)
Largura do trecho (m)	15,4 ($\pm 3,4$)	11,8 ($\pm 7,1$)	90,5 ($\pm 13,7$)	313,7 ($\pm 17,4$)	247,8 ($\pm 15,8$)	25,6 ($\pm 5,0$)
Qualidade da água						
Temperatura da água (°C)	30,4 ($\pm 3,3$)	31,4 ($\pm 2,1$)	31,6 ($\pm 2,7$)	26,7 ($\pm 2,8$)	27,7 ($\pm 2,0$)	29,0 ($\pm 0,1$)
Oxigénio dissolvido (mg/L)	4,9 ($\pm 1,6$)	6,0 ($\pm 0,5$)	7,1 ($\pm 2,5$)	6,1 ($\pm 3,3$)	5,1 ($\pm 2,3$)	5,2 ($\pm 0,3$)
Transparência (cm)	42,3 ($\pm 15,6$)	30,8 ($\pm 16,4$)	67,4 ($\pm 16,3$)	42,1 ($\pm 13,7$)	60,8 ($\pm 20,7$)	37,4 ($\pm 10,9$)
Composição do habitat (%)						
Cobertura de macrófitas	0.0	2.1 ($\pm 4,2$)	35,7 ($\pm 21,1$)	12,8 ($\pm 22,6$)	0.0	0.0
Capim na margem	5,8 ($\pm 11,7$)	11,1 ($\pm 10,4$)	2,1 ($\pm 4,0$)	5,2 ($\pm 6,5$)	34,5 ($\pm 22,3$)	0,7 ($\pm 1,2$)
Vegetação submersa	0,8 ($\pm 1,5$)	2,5 ($\pm 5,0$)	4,2 ($\pm 8,3$)	7,3 ($\pm 12,5$)	11,2 ($\pm 17,4$)	0.0
Vegetação marginal	17,1 ($\pm 15,5$)	0.0	10,4 ($\pm 15,8$)	0,9 ($\pm 1,6$)	0.0	0.0
Serapilheira	1,5 ($\pm 0,6$)	0,4 ($\pm 0,4$)	6,5 ($\pm 11,5$)	0.0	1,9 ($\pm 2,1$)	0,4 ($\pm 0,1$)
Alga	21,4 ($\pm 2,9$)	5,1 ($\pm 3,6$)	17,7 ($\pm 11,3$)	17,3 ($\pm 3,1$)	7,8 ($\pm 5,5$)	0,1 ($\pm 0,1$)
Massas de raízes	2,3 ($\pm 2,3$)	0.0	0.0	0.0	0.0	0.0
Galhos	4,7 ($\pm 1,3$)	2,6 ($\pm 1,3$)	7,2 ($\pm 2,1$)	0,9 ($\pm 0,7$)	15,5 ($\pm 0,4$)	1,1 ($\pm 0,8$)
Composição do substrato (%)						
Lama	17,8 ($\pm 21,7$)	66,4 ($\pm 22,5$)	48,1 ($\pm 32,2$)	94,4 ($\pm 4,6$)	52,7 ($\pm 32,1$)	40,5 ($\pm 7,7$)
Areia	60,0 ($\pm 27,5$)	23,7 ($\pm 16,2$)	45,5 ($\pm 36,1$)	3,9 ($\pm 2,0$)	45,1 ($\pm 33,0$)	55,6 ($\pm 7,6$)
Cascalho	9,2 ($\pm 0,0$)	0.0	4,3 ($\pm 0,7$)	1,7 ($\pm 0,4$)	0.0	2,7 ($\pm 1,9$)
Pedras	10,5 ($\pm 9,9$)	6,3 ($\pm 9,2$)	0.0	0.0	1,0 ($\pm 1,2$)	0,4 ($\pm 0,5$)
Rochas	2,5 ($\pm 5,0$)	2,8 ($\pm 3,9$)	0,0 ($\pm 0,0$)	0.0	1,3 ($\pm 2,5$)	0,8 ($\pm 0,7$)
Base rochosa	0.0	0,8 ($\pm 1,7$)	2,1 ($\pm 4,2$)	0.0	0.0	0.0

	Eixo 1	Eixo 2	Eixo 3
Autovalores	0.744	0.677	0.637
Teste de Monte Carlo	0.005	0.001	0.002
Variância nos dados das espécies			
% variância explicada	13,6	12,4	11,6
Correlações entre espécies e meio ambiente	1.000	1.000	1.000
Teste de Monte Carlo	0.001	0.001	0.001
Correlações intra-definidas			
Morfologia			
Comprimento do rio	-0,380	-0,498	0,304
Altitude	-0,377	-0,772	-0,172
Velocidade da água	0,318	0,252	0,456
Inclinação do banco	-0,712	0,266	-0,396
Profundidade litorânea	-0,117	0,709	0,378
Largura	-0,054	-0,674	-0,443
Qualidade da água			
Temperatura	0,238	0,586	-0,014
Composição do habitat			
Macrófitos	0,383	0,138	-0,417
Escombros	0,469	0,044	-0,022
Composição de substrato			
Areia	-0,294	0,487	-0,179
Cascalho	-0,347	0,083	-0,556





Referências

Chapter 5

Base de dados II - Habitat

Estrutura do Habitat: Variáveis ambientais no semiárido

Apresentação

Exemplo de conjunto de dados ecológicos do Programa de Pesquisa em Biodiversidade (PPBio) do Semiárido. Ano de 2006. Estrutura do habitat¹

Autor:

- Prof. Elvio S. F. Medeiros
- Laboratório de Ecologia
- Universidade Estadual da Paraíba
- Campus V, João Pessoa, PB

Para entender a distribuição das espécies de peixes e seu uso de habitat, uma série de variáveis ambientais foram avaliadas como preditores da composição e riqueza da assembleia de peixes em sistemas aquáticos tropicais semiáridos. Nós pesquisamos a composição de elementos físicos, químicos e da estrutura do habitat associados a assembleias de peixes em sistemas aquáticos semiáridos e estabelecemos seu grau de associação. Os locais consistiam em trechos de riachos com fluxo de água superficial, poças temporárias isoladas e reservatórios artificiais (açudes). A amostragem de elementos do habitat foi feita associada a amostragem da comunidade de peixes e zooplâncton em quatro ocasiões durante as estações chuvosa (abril e junho de 2006) e seca (setembro e dezembro de 2006).

Texto modificado de MEDEIROS; SILVA; RAMOS (2008): Application of catchment- and local-scale variables for aquatic habitat characterization and assessment in the Brazilian semi-arid region. Neotropical Biology and Conservation, 2008, vol. 3, n. 1, p. 13-20. <https://revistas.unisinos.br/index.php/>

¹Projeto financiado pelo CNPq/UEPB/DCR Proc.350082/2006-5, UEPB/FAPESQ Proc.68.0006/2006.0 e Projeto de Pesquisa em Biodiversidade do Semiárido (PPBio SemiÁrido).

[neotropical/article/view/5440](#). Tradução: Google Translator.

Elvio Sergio Figueredo Medeiros^{1,4}, Marcio Joaquim da Silva², e Robson Tamar Costa Ramos^{1,3}

¹ Grupo Ecologia de Rios do Semiárido, Universidade Estadual da Paraíba, Depto. de Biologia. Campus V. CEP 58070-450 João Pessoa - PB. Brazil. ² Universidade Federal do Rio Grande do Norte, Depto. de Biologia, Centro de Biociências, CEP 59078-900 – Natal - RN. ³ Laboratório de Sistemática e Morfologia de Peixes, Universidade Federal da Paraíba, Depto. de Sistemática e Ecologia, CCEN, CEP 58059-900, João Pessoa - PB.

- Autor correspondente: Dr. Elvio Medeiros <https://orcid.org/0000-0002-7472-8147>. Universidade Estadual da Paraíba, João Pessoa - Paraíba – Brasil - CEP 58.070-450. E-mail: elviomedeiros@servidor.uepb.edu.br

⁴Os autores agradecem a Telton Ramos e Virginia Diniz (UFPB) pela assistência em campo e ao CNPq/UEPB/DCR pela bolsa concedida a ESFM (350082/2006-5). Esta pesquisa contou com o apoio da UEPB/FAPESQ (68.0006/2006.0) e do Projeto de Pesquisa em Biodiversidade do Semiárido (PPBio Semi-Árido).

RESUMO

O estado do habitat físico é influenciado por fatores que atuam em diversas escalas, como a geomorfologia, o clima, o regime hidrológico, o uso da terra e a qualidade da água. Este trabalho quantifica a diversidade e a disponibilidade do habitat físico em sistemas aquáticos naturais e artificiais no semiárido do Brasil, com o objetivo de levantar variáveis físicas em escala local e de bacia de drenagem, e analisar sua importância na composição de uma estrutura básica para a caracterização e avaliação desses ambientes. Este estudo foi desenvolvido nas regiões de Seridó/Borborema e Buíque/Vale do Ipojuca. Estas áreas são consideradas de extrema importância biológica e são identificadas como prioritárias para a conservação da Caatinga. Os resultados mostram que o hábitat aquático no semiárido brasileiro é diverso e dinâmico, compreendendo um conjunto amplo de elementos disponíveis para a colonização pelos organismos. Os rios mostraram uma gama de elementos do hábitat marginal e da composição do substrato igual ou superior àquela dos reservatórios. Além disso, a composição do hábitat variou entre os dois tipos de ambiente estudados (rios/reservatórios) e com as estações do ano (seca/chuvosa). Os resultados apresentados têm implicações importantes para a conservação dos ambientes aquáticos do semiárido. Tendo em vista que o habitat físico é a estrutura básica para a colonização por parte dos organismos aquáticos, os mecanismos potenciais que mantêm a biodiversidade aquática ocorrem nos vários níveis da bacia de drenagem. Portanto, é fundamental identificar os componentes, ao longo dos sistemas ripários, que são vitais na manutenção da sua integridade biológica.

Palavras-chave: rios intermitentes, reservatórios, conservação, composição de

substratos

5.1 Introdução

Córregos intermitentes no semiárido brasileiro são características paisagísticas distintivas, existindo como cursos d'água secos durante a maior parte do ano. Maltchik e Medeiros (2006) reconheceram que os extremos de inundação e seca são impulsionadores de importantes processos que mantêm a diversidade dentro desses sistemas. Portanto, as medidas de conservação no semiárido brasileiro devem incluir a manutenção do regime de fluxo natural dos sistemas aquáticos, para garantir a sobrevivência das espécies em longo prazo (Maltchik e Medeiros, 2006). No entanto, as medidas de conservação também devem levar em conta a composição e diversidade do habitat aquático disponível, pois o habitat físico é o quadro de colonização pela fauna aquática (eg Martin-Smith, 1998). Além disso, o estado deste espaço vital influenciará a estrutura biótica e a organização dentro dos sistemas aquáticos (Mugodo et al., 2006). O estado do habitat físico, nomeadamente composição e diversidade, é influenciado por factores que operam a várias escalas espaciais e temporais (Boys e Thoms, 2006). Ao nível da bacia, a geomorfologia e o clima influenciarão o habitat à escala de alcance, afectando a hidrologia, sedimentação, aportes de nutrientes e morfologia do canal (Davies et al., 2000; Mugodo et al., 2006). No nível local, o uso e manejo da terra também influenciarão o habitat em escala de alcance do córrego (Richards et al., 1996).

Os métodos de avaliação do habitat físico disponíveis para os organismos aquáticos fornecem ferramentas importantes para vários aspectos da gestão e conservação do rio e, portanto, os procedimentos usados para avaliar o habitat devem ser ecologicamente e geomorfologicamente significativos (Thomson et al., 2001). Apesar disso, métodos de avaliação de habitats precisam ser desenvolvidos e testados em sistemas aquáticos semiáridos do Brasil, pois não são tão bem desenvolvidos quanto os métodos usados para avaliar outros aspectos da condição dos rios, como a qualidade da água. A avaliação do habitat é de grande utilidade para prever a distribuição potencial de espécies-chave, como peixes (Boys e Thoms, 2006) e invertebrados (Richards et al., 1997), com base em suas necessidades de habitat. Além disso, o habitat observado e potencial (ou previsto) pode ser comparado com os requisitos de habitat de uma espécie para avaliar a necessidade de manejo do habitat e a saúde geral do ecossistema com base nas necessidades da biota aquática (Maddock, 1999).

Este trabalho mede a diversidade e disponibilidade do habitat físico em sistemas aquáticos naturais e artificiais no nordeste do Brasil. Este artigo tem como objetivo levantar variáveis físicas em escala local e de captação e avaliar sua importância como marco básico para caracterização e avaliação de habitats aquáticos em duas áreas do semiárido brasileiro.

5.1.1 Desenho Amostral

Este estudo foi realizado na região semiárida brasileira, em bacias hidrográficas que fluem através de uma floresta aberta arbustiva seca decidual (a “Caatinga”). A amplitude térmica é baixa na área de estudo, com médias variando de aproximadamente 25 a 30 °C, e a precipitação média anual varia entre 600 e 1100 mm. As altitudes variam entre 100 e 1000 m (Figura 5.1 e Figura 5.2).

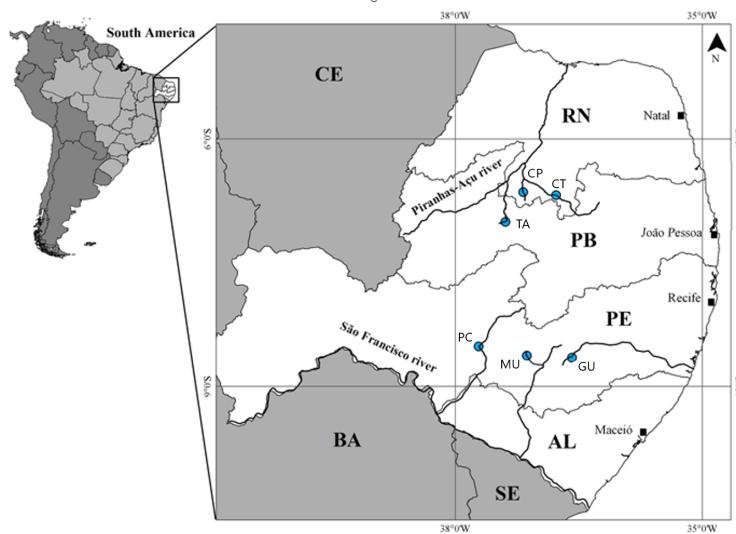


Figure 5.1: Área de estudo mostrando os estados do Rio Grande do Norte (RN), Paraíba (PB), Pernambuco (PE) e Alagoas (AL), principais sistemas fluviais e pontos de amostragem na região semiárida do Brasil. TA, açude Recanto; CP, riacho Cipó; CT, riacho Catureré; PC, riacho Poço da Cruz; MU, açude Mulungu e GU, açude Gurjão.

Duas regiões de importância biológica (**Seridó, S e Buíque, B**) foram colhidas e nelas, **seis locais** ou sítios de coleta (TA, CP, CT, PC, MU, e GU) foram selecionados para representar ambientes Unidades Amostrais de ambientes temporários naturais e artificiais típicos (Figura 5.1 e Figura 5.4).

Os locais consistiam em trechos de **riachos (-R)** com fluxo de água superficial (durante a estação chuvosa) ou poças temporárias isoladas (durante a estação seca) e **reservatórios ou açudes (-A)** artificiais criados a partir do represamento de riachos. A amostragem foi realizada durante o ano de 2006 em **quatro ocasiões** durante as estações chuvosa (**abril, 1 e junho, 2**) e seca (**setembro, 3 e dezembro, 4**) (Figura 5.1 e Figura 5.4).

Table 1. General characteristics of the study sites. *A-priori* criteria for grouping habitat structure in the DCA are indicated with a (*).

Sample sites	Area *	Catchment Basin	River	Stream site location	Habitat type *	Position
1 CT	Seridó/Borborema	Piranhas-Açu	Seridó	Seridó River	Stream	06° 36' 30.24" S 36° 46' 57.18" W
2 CP	Seridó/Borborema	Piranhas-Açu	Sabugi	Cipó Stream	Stream	06° 38' 39.42" S 37° 10' 30.84" W
3 TA	Seridó/Borborema	Piranhas-Açu	Espinharas	Recanto Stream	Reservoir	07° 00' 18.06" S 37° 23' 28.20" W
4 PC	Buique/Vale do Ipojuca	São Francisco	Moxotó	Escama Peixe Stream	Stream	08° 30' 55.74" S 37° 42' 58.44" W
5 MU	Buique/Vale do Ipojuca	São Francisco	Ipanema	Mulungu Stream	Reservoir	08° 37' 35.94" S 37° 07' 44.40" W
6 GU	Buique/Vale do Ipojuca	Una	Una	Salobro Stream	Reservoir	08° 39' 04.26" S 36° 34' 57.24" W

Figure 5.2: Características gerais dos locais de estudo. Os critérios a priori para agrupar a estrutura do habitat no DCA são indicados com um (*).

5.1.2 Sobre os dados do PPBio

A planilha ppbio contém os dados das variáveis ambientais em diferentes unidades amostrais (UA's) (Figura 5.3) (Veja Programa de Pesquisa em Biodiversidade – PPBio). Essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados (não foram relativizados ou transformados). Outros tipos de arquivos existem para os dados do PPBio (3.2), que fazem parte de um estudo mais amplo sobre ecologia de rios do semiárido.

s.ind	s.land	s.smgrev	s.tgrev	s.cobbles	s.rocks	s.bedrock	m.elev	m.river	m.stream	m.disttou	m.distmou	m.distlou	m.maxslope	m.maxdist	m.habitdebt	m.width	a.veloc	a.temp	a.dio	a.hrange
S-R-CP1	65	30	0	1,6666667	3,3333333	226	163,2	163,2	84,45	217,15	30	106	81,33333	19,64	0,1666667	32,9	6,51	46		
S-R-CP1	16,6666667	70	5	5	3,3333333	0	169	83,05	13,95	6,15	185,55	60	60	22,66667	17,24	0,159151	35,2	8,6888888	26	
S-R-CP1	83,3333333	8,3333333	1,6666667	1	0	8,3333333	270	163,2	163,2	84,45	217,15	30	128	14,66667	19,64	0,1666667	32,9	6,51	46	
S-R-CP2	40	40	0	0	20	0	226	163,2	163,2	84,45	217,15	30	105	67	16,1	0,125	32	5,375	44	
S-R-CP2	0,6666667	87,66666	3,3333333	3,3333333	5	0	169	83,05	13,95	6,15	185,55	60	68	32,66667	18,47	0	19	5,015	33	
S-R-CP3	1,6666667	87,66666	3,3333333	3,3333333	0	0	270	163,2	163,2	84,45	217,15	30	110	12,66667	19,64	0,1666667	32,9	6,51	46	
S-R-CP3	65,55555	23,11111	0	0	3	8,3333333	0	226	163,2	163,2	84,45	217,15	30	110	32,33333	6,2	0	28,26667	6	17,33333
S-R-CP3	48,75	23,5	10	10	8,75	0	169	83,05	13,95	6,15	185,55	60	79	32,66667	15,1	0	29,66667	5	50,33333	
S-R-CP4	46,75	40	6,6666667	6,6666667	0	0	270	163,2	163,2	84,45	217,15	30	104	12,66667	19,64	0,1666667	32,9	6,51	46	
S-R-CP4	95	1,8333333	0	0	0,3333333	2,8333333	0	226	163,2	163,2	84,45	217,15	30	74	32,33333	5,4	0	32,6	6,51	46
S-R-TA1	5	60	0	25	10	0	169	83,05	13,95	6,15	185,55	60	64	45,33333	10,7	0	27,6	4,9	60	
S-R-TA1	55,3333333	38,7142857	0	0	0	0	270	163,2	163,2	84,45	217,15	30	67	30,33333	72,2	0	29,33333	9,4333333	67	
S-A-AM1	55,3333333	65	0	0	0	0	73	212,6667	19,83333	73,85833	190,83333	30	125	10,66667	20,66667	0	29,83333	5,6666667	48	
S-A-GU1	96,66666	3,3333333	0	0	1,6666667	0	73	212,6667	19,83333	73,85833	190,83333	30	60	18,66667	33	0	29,33333	5,1333333	29,66667	
S-A-PG1	39	56	3	1	1	0	402	196,8333	32,375	11,41667	401,6667	90	110	49,33333	29,6	0	19	5,635	50	
S-A-PG2	77	77	0	24	0	0	73	212,6667	19,83333	73,85833	190,83333	30	110	12,66667	20,66667	0	29	1,625	43	
S-A-GU2	98	2	0	0	0	0	73	212,6667	32,33333	10,16667	203,8333	30	69	7	321	0	19	1,85	55	
S-A-PG3	33,66667	63	3,3333333	0	0	0	402	196,8333	32,375	11,41667	401,6667	90	89	50	27,3	0	19	5	32,33333	
S-A-GU3	98	35	0	0	0	0	73	212,6667	19,83333	73,85833	190,83333	30	110	25,66667	20,66667	0	29	5,75	48	
S-A-GU3	87,77778	6,6666667	3,3333333	2,222222	0	0	73	212,6667	32,33333	10,16667	203,83333	30	68	8,833333	314,2	0	24	8,8	51,66667	
S-B-PC1	48,875	47,875	1,875	0	0,125	1,25	0	402	196,8333	32,375	11,41667	401,6667	90	95	52,83333	20	0	28,85	5,1	30
S-B-MU4	91,77778	3,222222	0	0	5	0	725	214,02	19,83333	73,85833	290,8767	30	112	41,33333	239	0	15,95	7,5	89	
S-B-GU4	95,66666	8,4166667	1,3333333	0,083333	0	0	713	212,6667	31,33333	10,16667	203,83333	30	69	4,666667	289,5	0	24,7	8,75	36	

Figure 5.3: Parte da planilha de dados brutos do PPBio para o habitat.

5.2 Codificação para variáveis

Os arquivos da base de dados do projeto são fornecidos em formato Excel (.xlsx). Por exemplo, o arquivo ppbio06*-habitat.xlsx, traz os dados brutos que serão usados nas análises. A matriz de dados brutos contém mais de 20 localidades (n=linhas ou objetos) em estações do ano diferentes, e cerca de 30 variáveis ambientais (m=colunas ou atributos), antes de qualquer modificação. Portanto é uma matriz bruta. Os valores incluem diversos tipos de variáveis medidas em unidades diferentes (% , °C, mg/L, etc), e apresentam uma alta amplitude de variação, portanto, a aplicação de uma transformação é recomendado (Figura 5.3). Nos nomes dos objetos e dos atributos são codificados de acordo com a tabela mostrada na Figura 5.4.

Variáveis da estrutura do habitat (Atributos)	Sítios e períodos de amostragem (Objetos)												
<p>Habitat marginal (% de cobertura)</p> <ul style="list-style-type: none"> Grama litorânea Macrófitas Algas aderidas (a um substrato) Algas filamentosas Vegetação pendente (encobrindo a margem) Vegetação (terrestre) submersa Galhos pequenos (<2 cm de diâmetro) Galhos grandes (>2 cm de diâmetro) Liteira ou folhig. (submerso) Massas de raízes <p>Composição do substrato (% de cobertura)</p> <ul style="list-style-type: none"> Lama (ou siltete) Arcia Seixos Cascalho pequeno (<2 cm) Cascalho grande (>2 cm) Rochas Rocha mãe <p>Qualidade da água</p> <ul style="list-style-type: none"> Velocidade da água (m/s) Temperatura (°C) Oxigênio dissolvido (mg/L) Transparência (cm) de Sechi <p>Agrupamentos a-priori</p> <ul style="list-style-type: none"> Por tipo de ambiente: Rio / Reservatórios Por área de estudo: Seridó / Buique Por estação: Chuvosa / Seca <p>Escalas das variáveis</p> <p>Escala de bacia e morfologia (escala regional)</p> <ul style="list-style-type: none"> Comprimento total da bacia (km) Comprimento do riacho do sítio amostrado (km) Distância do sítio para a nascente do rio (km) Distância do sítio para a foz do rio (km) Elevação (= altitude) (m.a.n.m.) <p>Escala de sítio e morfologia (escala local)</p> <ul style="list-style-type: none"> Declive da margem (graus) Largura do trecho amostrado (m) Profundidade na margem do trecho amostrado (cm) Profundidade máxima no trecho amostrado (cm) <p>Escala de sítio e qualidade da água (escala local)</p> <ul style="list-style-type: none"> Velocidade da água (m/s) Oxigênio Dissolvido (mg/L) Temperatura (°C) Transparência da água (cm) 	<p>Nos códigos dos sítios está o desenho amostral. Duas áreas, ao norte o Seridó (S) e ao sul Buique (B). Em cada área existem dois tipos de ambiente, açude (A) e rio (R). Ao norte estão três sítios, TA, CP e CT, e ao sul três sítios, PC, MU e GU. Em cada sítio há quatro amostragens por ano, trimestre 1, 2, 3 e 4.</p> <p>Resumo:</p> <ul style="list-style-type: none"> S: Seridó B: Buique A: Açude R: Rio <table style="margin-left: 20px;"> <tr><td>TA: Açude Recanto</td><td>(RE)</td></tr> <tr><td>CP: Riacho Cipó</td><td>(CI)</td></tr> <tr><td>CT: Riacho Catureré</td><td>(SE)</td></tr> <tr><td>PC: Riacho Poço da Cruz</td><td>(EP)</td></tr> <tr><td>MU: Açude Mulungu</td><td>(MU)</td></tr> <tr><td>GU: Açude Gurjão</td><td>(SA)</td></tr> </table> <p>Exemplo:</p> <p>A amostra S-R-CP-1 representa, a região Seridó (S), um ambiente de rio (R), o sítio riacho Cipó (CP), no primeiro trimestre do ano (1).</p>	TA: Açude Recanto	(RE)	CP: Riacho Cipó	(CI)	CT: Riacho Catureré	(SE)	PC: Riacho Poço da Cruz	(EP)	MU: Açude Mulungu	(MU)	GU: Açude Gurjão	(SA)
TA: Açude Recanto	(RE)												
CP: Riacho Cipó	(CI)												
CT: Riacho Catureré	(SE)												
PC: Riacho Poço da Cruz	(EP)												
MU: Açude Mulungu	(MU)												
GU: Açude Gurjão	(SA)												

Figure 5.4: Codificação para as variáveis ambientais descritoras da estrutura do habitat (atributos) e os sítios e períodos de amostragem (objetos).

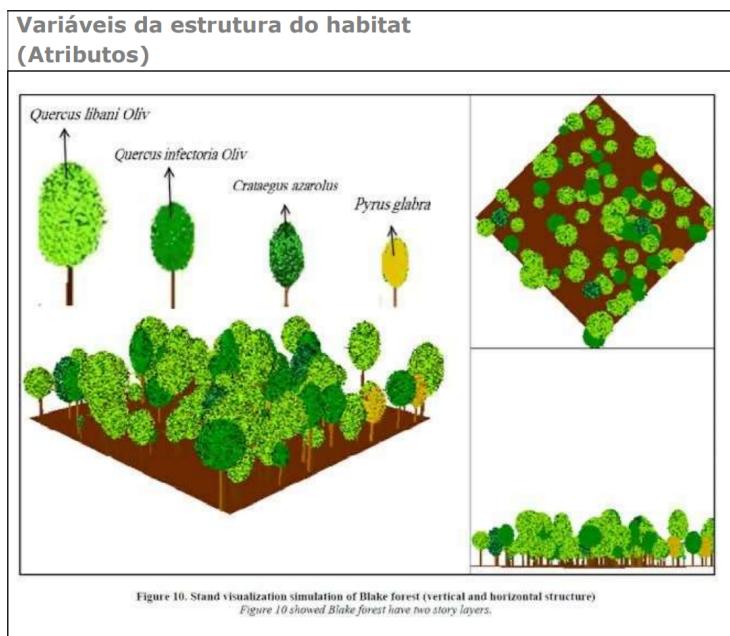


Figure 5.5: Imagem representativa da codificação das variáveis, para a medição da estrutura do habitat (atributos).

5.3 Área de estudo e amostragem dos dados

Este estudo foi realizado em duas áreas diferentes do semiárido brasileiro: Seridó/Borborema e Buíque/Vale do Ipojuca (conforme Tabarelli e Silva, 2003) (Tabela 1; Figura 1). Essas áreas são classificadas como de extrema importância biológica e foram identificadas como áreas prioritárias para conservação da biodiversidade na Caatinga por Silva et al. (2003), por apresentarem alta diversidade de espécies e serem ricas em endemismos. A área do Seridó está localizada no sul do Rio Grande do Norte (RN) e norte da Paraíba (PB) entre Patos e Caicó (Figura 5.1). A temperatura média anual é de 30,7°C, com máxima média mensal em outubro (31,0°C) e mínima média em fevereiro (29,3°C). A precipitação concentra-se entre janeiro e abril, com 350 a 800 mm por ano e média anual de 600 mm (Amorim et al., 2005). A altitude no Seridó varia entre 100 e 800 m (Governo do Estado da Paraíba, 1985). A área de Buíque está localizada no entorno da cidade de Buíque no centro de Pernambuco (PE) (Figura 5.1. A temperatura média anual e a precipitação são 25°C e 1095,9 mm, respectivamente. As chuvas concentram-se entre abril e junho. Altitude varia entre 800 e 1000 m (Rodal et al., 1998).

Dentro de cada área de estudo, três locais foram selecionados para representar ambientes típicos artificiais e naturais temporários e semipermanentes (Tabela 1). Os locais consistiam em trechos de córregos, geralmente de 100 a 500 m de comprimento, e reservatórios artificiais criados a partir de represamento de córregos. A amostragem foi realizada durante um ano em quatro ocasiões durante as estações chuvosa (abril e junho de 2006) e seca (setembro e dezembro de 2006). Variáveis de escala de alcance de captação e córrego foram selecionadas com base nas características locais e em Pusey et al. (2004) (entre outros, Richards et al., 1996; Davies et al., 2000; Thomson et al., 2001; Mugodo et al., 2006). As variáveis de escala de captação foram quantificadas com base nas características topográficas gerais da área de estudo e nas características da bacia hidrográfica usando mapas topográficos 1:500.000, imagens de satélite e receptor GPS. As variáveis do local em escala de alcance apresentam características morfométricas e de qualidade (físico-químicas) da água. A estrutura do habitat compreendeu (i) elementos marginais subaquáticos e litorâneos (representando a abundância de elementos marginais de microhabitat) e (ii) a composição do substrato.

A qualidade da água em escala de alcance, a estrutura do habitat e algumas das características morfométricas foram estimadas em pontos de levantamento de um metro quadrado e distribuídas em categorias. Normalmente, 3 a 12 pontos de pesquisa aleatórios foram avaliados dentro de cada trecho de rio ou reservatório. A composição do substrato e os elementos do habitat foram estimados como sua contribuição proporcional ao perímetro molhado do local. Para a avaliação da morfologia do local, a largura foi medida com fita métrica para distâncias de até 100 m e receptor de satélite GPS para distâncias maiores. A profundidade foi medida com um bastão a distâncias aproximadamente equivalentes ao longo de um transecto para representar a profundidade do habitat (profundidade média

dos primeiros 3 m das margens) e a profundidade máxima. A velocidade da água foi medida usando o método de flutuação (Maitland, 1990). A inclinação da margem foi estimada visualmente e alocada em categorias de inclinação (< 30°, 30-60° e 60-90°). A temperatura da água (°C) e o oxigênio dissolvido (mg/l) foram medidos com um medidor de oxigênio (Lutron DO-5510), e a transparência (cm) foi medida com um disco Secchi.

5.4 Análises estatísticas

A variação na composição do habitat entre os locais e as ocasiões de amostragem foi investigada usando a Análise de Correspondência Descendente (DCA) dos dados padronizados transformados em arcsine-squareroot. O Multi-Response Permutation Procedure (MRPP) (Biondini et al., 1985; McCune e Grace, 2002) foi usado para testar diferenças na composição de habitats entre as duas áreas de estudo (Seridó e Buíque), tipos de habitats (córrego e reservatório) e estações do ano (chuvoso e seco). Para todas as análises de MRPP, a concordância dentro do grupo corrigida pela chance (A) é apresentada como uma medida do grau de homogeneidade dentro do grupo, em comparação com a expectativa aleatória. Onde o MRPP detectou diferenças significativas, uma análise adicional foi realizada para revelar quais elementos particulares do habitat contribuíram significativamente como a fonte da diferença na composição do habitat, usando a Análise de Espécies Indicadoras de McCune e Mefford (1999). Os Valores Indicadores (IV) foram calculados pelo método de Dufrene e Legendre (1997). Estes foram testados para significância estatística ($p < 0,05$) usando uma técnica de Monte Carlo com 1000 corridas. A influência das variáveis de alcance e escala de captação na estrutura do habitat foi avaliada usando a Análise de Correspondência Canônica (CCA) segundo McCune e Grace (2002). Os dados foram centralizados e normalizados e testados com randomização de Monte Carlo (100 corridas). Todas as estatísticas foram realizadas em PC-ORD 4.27 (McCune e Mefford, 1999).

5.5 Resultados

A estrutura do habitat para todos os locais de estudo consistiu geralmente em plantas C4 crescendo nas margens do litoral (nomeadamente gramíneas), macrófitas aquáticas, incluindo plantas flutuantes (geralmente *Salvinia* sp., *Pistia* sp. e *Azolla* sp.), plantas emergentes (*Nymphaea* spp.) e plantas submersas (*Ceratophyllum* sp. e *Egeria* sp.) e algas (aderidas ao substrato e filamentosas) (Figura 5.6). Vegetação ciliar pendendo, restos lenhosos, serapilheira e massas de raízes também estiveram presentes nos locais estudados.

A composição do substrato foi composta principalmente de lama e areia, com menor contribuição de seixos, pedras e cascalho (Figura 5.6). A diversidade de estruturas de habitat foi geralmente maior nos locais dos rios, na área do Seridó e durante a estação chuvosa, apesar de uma maior riqueza de elementos

Table 2. Percentages (\pm standard deviation) of the habitat elements of both marginal elements and substrate composition occupying the marginal wetted perimeter of the study sites and average water quality parameters.

Habitat elements	Average(SD)	Minimum-maximum
Marginal habitat		
Littoral grass	10.3 (± 15.8)	0-54.0
Macrophytes	8.8 (± 17.7)	0-54.8
Attached algae	6.5 (± 14.2)	0-50.0
Filamentous algae	5.6 (± 10.5)	0-33.3
Overhanging vegetation	4.9 (± 10.7)	0-33.3
Submerged vegetation	4.5 (± 9.6)	0-36.6
Small debris (<2 cm diameter)	3.5 (± 3.3)	0-10.0
Large debris (>2 cm diameter)	2.0 (± 4.5)	0-20.0
Leaf litter	1.8 (± 4.9)	0-23.7
Root masses	0.4 (± 1.2)	0-5.0
Substrate composition		
Mud	53.9 (± 31.9)	0.7-98.0
Sand	38.2 (± 29.5)	1.8-95.0
Cobbles	3.1 (± 6.5)	0-25.0
Small gravel (<2 cm)	1.8 (± 2.6)	0-10.0
Rocks	1.2 (± 2.8)	0-10.0
Large gravel (>2 cm)	1.2 (± 2.6)	0-10.0
Bedrock	0.5 (± 1.8)	0-8.3
Water quality		
Temperature (°C)	29.5 (± 2.9)	24.0-35.2
Dissolved oxygen (mg/l)	5.8 (± 2.0)	1.8-9.4
Transparency (cm)	47.2 (± 19.5)	16.0-90.0

Figure 5.6: Percentagens (\pm desvio padrão) dos elementos do habitat de ambos os elementos marginais e composição do substrato que ocupam o perímetro molhado marginal dos locais de estudo e parâmetros médios de qualidade da água.

de habitat nos locais dos reservatórios (Figura 5.7). A composição do substrato também foi mais diversa e rica nos locais do rio e na área do Seridó, enquanto o substrato foi mais rico e diversificado durante a estação seca (Figura 5.7). A análise de correspondência sem tendência (Figura 5.8) mostrou que as diferenças na estrutura do habitat foram significativas apenas entre os locais de riachos e reservatórios (MRPP, $A = 0,06$, $p = 0,03$) e estações (MRPP, $A = 0,06$, $p = 0,03$).

Table 3. Diversity and richness of habitat elements in study sites across habitat types, study area and seasons.

Habitat elements	A-priori groups	Richness (R)	Shannon's Diversity (H)
Marginal habitat			
Habitat type	River	4.4 (± 2.3)	0.942 (± 0.501)
	Reservoir	4.7 (± 1.6)	0.937 (± 0.397)
Area	Seridó	5.2 (± 2.2)	1.040 (± 0.539)
	Buique	3.8 (± 1.4)	0.829 (± 0.283)
Season	Wet	5.3 (± 2.1)	1.083 (± 0.463)
	Dry	3.8 (± 1.6)	0.808 (± 0.389)
Substrate composition			
Habitat type	River	4.3 (± 0.8)	0.857 (± 0.289)
	Reservoir	3.0 (± 0.9)	0.497 (± 0.300)
Area	Seridó	3.9 (± 0.9)	0.801 (± 0.348)
	Buique	3.3 (± 1.1)	0.525 (± 0.283)
Season	Wet	3.5 (± 1.2)	0.607 (± 0.333)
	Dry	3.8 (± 0.9)	0.727 (± 0.354)

Figure 5.7: Diversidade e riqueza de elementos do habitat em locais de estudo em todos os tipos de habitat, área de estudo e estações do ano.

Diferenças na estrutura do habitat entre as duas áreas de estudo não foram significativas (MRPP, $A = 0,01$, $p = 0,26$). A análise do valor indicador mostrou que os elementos importantes que separam esses grupos foram macrófitas para reservatórios ($IV = 53,2$, $p = 0,01$) e areia e pedras para locais fluviais ($IV = 61,8$, $p = 0,02$ e $IV = 82,6$, $p = 0,002$, respectivamente). Importantes características de habitat que separam as estações foram vegetação submersa ($IV = 48,4$, $0,01$) e vegetação pendente ($IV = 48,9$, $p = 0,02$) para a estação chuvosa. Nenhum elemento do substrato foi significativamente importante na separação dos locais de acordo com as estações do ano (Figura 5.8).

A análise de correspondência canônica mostrou que as variáveis de alcance e escala de captação não estavam fortemente correlacionadas, exceto para a distância do local à nascente do rio e comprimento do córrego da localização do local (correlação de Pearson = 0,965), elevação e comprimento do rio principal (Correlação de Pearson = 0,853) e elevação e largura do local (correlação de Pearson = 0,906). A variância total dos dados (“inércia”) foi de 1,19. A variância total explicada pela CCA foi de 46,2%, sendo a maior parte dessa variação explicada pelo primeiro eixo (18,4%) (Figura 5.9. No entanto, os eixos 2 e 3 não podem ser desconsiderados, pois também explicaram parte substancial da variação do conjunto de dados. Os resultados da análise mostram uma relação

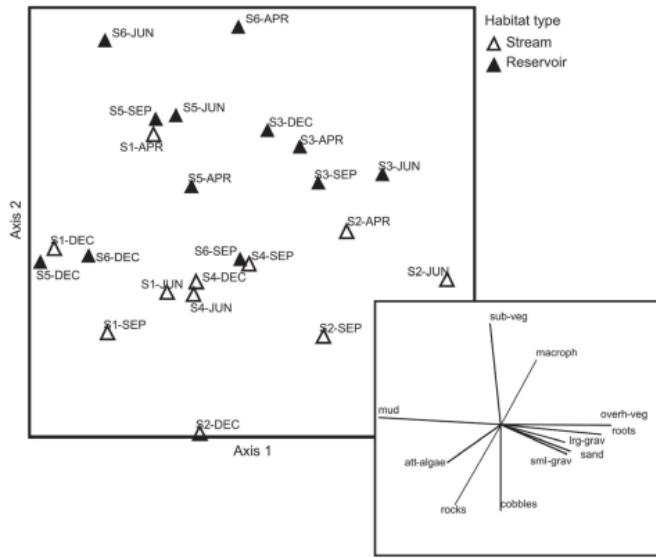


Figure 2. DCA results for habitat composition across the study sites. Insert box shows habitat elements correlated ($r^2>0.2$) with sampling sites/occasions in ordination space (denoted by vectors). The direction and length of vectors indicate strength of correlation. See Table 2 for full names of habitat elements. Sites (S1-S6) are coded as per Table 1 and sampling occasion. Abbreviations: APR, April; JUN, June; SEP, September; DEC, December.

Figure 5.8: Resultados da DCA para a composição do habitat nos locais de estudo. A caixa de inserção mostra elementos de habitat correlacionados ($r^2>0,2$) com locais/ocasiões de amostragem no espaço de ordenação (indicados por vetores). A direção e o comprimento dos vetores indicam a força da correlação. Consulte a Tabela 2 para obter os nomes completos dos elementos do habitat. Os locais (S1-S6) são codificados de acordo com a Tabela 1 e a ocasião da amostragem. Abreviaturas: APR, abril; JUN, junho; SET, setembro; DEZ, dezembro.

significativa entre os dados de estrutura do habitat e as variáveis de alcance e escala de captação, com o autovalor do primeiro eixo muito alto para ser esperado por acaso (Figura 5.9, valores de p para autovalor e correlação para o primeiro eixo = 0,01). O primeiro eixo foi fortemente correlacionado com variáveis morfológicas da escala de captação, nomeadamente comprimento do rio principal, elevação e a distância altamente correlacionada do local à nascente do rio e comprimento do córrego da localização do local. O primeiro eixo também foi correlacionado com a inclinação da margem, uma variável morfológica da escala de alcance. Eixos 2 e 3 foram correlacionados com variáveis de escala de alcance de morfologia (largura do local, eixo 2) e qualidade da água (velocidade da água e oxigênio dissolvido, eixo 3) (Figura 5.9).

Table 4. Axis summary statistics of Canonical Correspondence Analysis for habitat structure and catchment- and reach-scale variables.

	Axis 1	Axis 2	Axis 3
Eigenvalue	0.220	0.179	0.153
Variance in species data			
% of variance explained	18.4	15.0	12.8
Cumulative % explained	18.4	33.4	46.2
Pearson Correlation, Spp-Env	0.990	0.935	0.948
Kendall (Rank) Corr., Spp-Env	0.905	0.771	0.771
Total variance ("inertia")	1.19		

Table 5. Correlations for catchment- and reach-scale variables with the ordination axes after Canonical Correspondence Analysis.

		Correlations			
Catchment scale morphology	Variable				
		Axis 1	Axis 2	Axis 3	
	River length	-0.792	-0.177	0.142	
	Length of stream of site location	-0.416	-0.120	-0.051	
	Distance of site from source	-0.465	0.085	-0.082	
	Distance of site from mouth	-0.266	0.021	-0.010	
	Elevation	-0.553	-0.300	0.251	
Reach scale morphology	Margin slope				
		0.595	0.159	-0.317	
Reach scale morphology		-0.295	-0.520	0.329	
		0.024	0.346	0.035	
		-0.154	-0.070	0.355	
Reach scale water quality	Velocity				
		0.132	0.047	0.423	
		-0.026	-0.238	-0.595	
		0.324	-0.080	0.320	
		0.098	-0.281	0.091	

Figure 5.9: Estatísticas de sumário dos eixos da Análise de Correspondência Canônica para a estrutura do habitat e variáveis de escala de captação e alcance. Correlações para variáveis de escala de captação e alcance com os eixos de ordenação após Análise de Correspondência Canônica.

A Figura 5.10 indica que o eixo 1 representa um gradiente de sítios com estrutura de habitat influenciada por variáveis de escala de captação e o eixo 2 representa sítios influenciados por variáveis de escala de alcance.

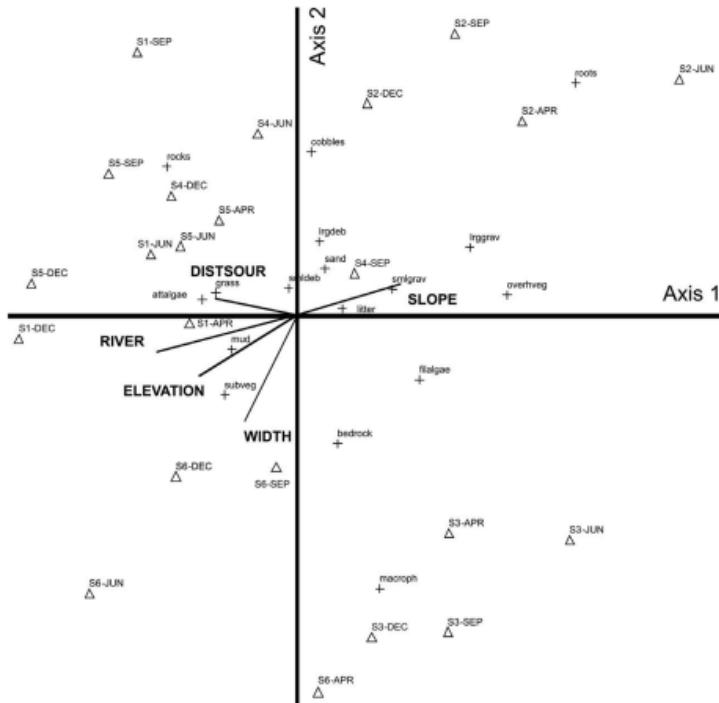


Figure 3. Biplot of ordination sites (Δ) (based on habitat structure) in the catchment- and reach-scale variables space, as defined by CCA. Vectors represent river length (RIVER), distance of site from source (DISTSOUR), elevation, margin slope (SLOPE) and site width (WIDTH). (+) indicates position of habitat elements in the ordination space.

Figure 5.10: Biplot de locais de ordenação (Δ) (com base na estrutura do habitat) na bacia e alcance- variáveis de escala espaço, conforme definido pelo CCA. Os vetores representam o comprimento do rio (RIVER), distância do local da fonte (DISTSOUR), elevação, inclinação da margem (SLOPE) e largura do local (WIDTH). (+) indica a posição dos elementos do habitat no espaço de ordenação.

5.6 Discussão

O presente estudo mostrou que o habitat aquático no semiárido brasileiro é diversificado e dinâmico, com uma gama de elementos de habitat disponíveis para colonização pela biota aquática. Sítios de riachos apresentaram uma composição de substrato e elementos de habitats marginais semelhante a maior quando comparados a reservatórios, e a composição do habitat variou com o tipo de habitat (rio/reservatório) e estações (seco/úmido). Mudanças sazonais nas características físicas de ambientes variáveis foram relatadas para regiões secas em outros lugares es (Mackey, 1991; Davis et al., 2002). Essas mudanças geralmente estão associadas a flutuações nos níveis de água e morfologia do habitat (Osborne et al., 1987; Medeiros, 2005). No presente estudo, as variações sazonais do nível da água e do regime hidrológico influenciaram a morfometria e a estrutura do habitat dos locais de estudo. Dadas as características lóticas dos locais de riachos, areia e seixos foram as características de habitat mais importantes neste tipo de habitat, pois espera-se que substratos finos como lama sejam realizados durante as inundações. A Figura 2 também dá indicação de que rochas, cascalho e algas anexadas eram elementos importantes de habitat em locais de riachos. Já os reservatórios foram caracterizados principalmente pela presença de macrófitas, vegetação submersa e, em menor grau, lama. O fato de as macrófitas aquáticas não terem sido elementos importantes na definição de habitat nos córregos intermitentes estudados está relacionado aos extremos de imundação durante o período úmido. Maltchik e Pedro (2001) e Pedro et al. (2006) constataram que a inundaçāo limitou a ocorrência de macrófitas aquáticas em riachos no semiárido brasileiro, onde as comunidades de macrófitas aquáticas sujeitas a inundaçāo apresentaram menor riqueza de espécies do que as comunidades sem inundaçāes. Os ciclos secos e úmidos em riachos semiáridos do nordeste do Brasil são caracterizados por extremos de inundações e secas e considerados como os mais importantes impulsionadores da estrutura da comunidade nesses sistemas (Maltchik e Medeiros, 2001; Maltchik e Florin, 2002). Os resultados do presente estudo mostram que os ciclos seco e úmido também desempenham um papel importante na estrutura do habitat, uma vez que a composição do habitat marginal e do substrato foram diferentes entre as estações. Curiosamente, o habitat marginal foi mais diverso durante a estação chuvosa do que durante a estação seca. Esse resultado discorda de outros estudos sobre estrutura de habitat em sistemas temporários ou semipermanentes (por exemplo, Kennard, 1995; Medeiros, 2005), que constataram que a riqueza e a diversidade de estruturas de habitat foram maiores durante a estação seca, quando a água recua níveis provocam a exposição de estruturas físicas anteriormente submersas, como galhos e troncos. Os dados apresentados neste estudo indicam um efeito oposto, pois a vegetação submersa e a vegetação pendente foram elementos característicos do habitat durante a estação chuvosa. Dado o aumento da área dos sistemas aquáticos durante o período úmido, a vegetação que outrora ocupou as margens secas do litoral de rios e reservatórios é inundada, agregando ainda mais complexidade ao habitat. Além disso, à medida que os riachos inundados atingem suas margens, o contato com a vegetação ciliar atinge seu máximo, aumentando a diversidade de estruturas.

tando ainda mais a complexidade do habitat (por exemplo, Pusey e Arthington, 2003). Como indicado pela alta importância da vegetação pendente durante a estação chuvosa, proximidade com a vegetação ciliar gera sombra, limitando o crescimento de macrófitas, mas acrescenta elementos ao habitat litorâneo como massas de raízes, serrapilheira e detritos. A composição do substrato, por outro lado, é mais provável de ser descoberta durante a estação seca, mais importante ainda nos riachos intermitentes, onde características como pedras e cascalho tendem a ser mais frequentes em trechos mais profundos.

Apesar das diferenças na diversidade e riqueza de elementos de habitat entre Seridó e Buíque, a estrutura de habitat não foi diferente entre as duas áreas de estudo. Como o desenho amostral não é equilibrado (dois reservatórios na área do Buíque e apenas um na área do Seridó), a diversidade e a riqueza de elementos do habitat tendem a ser maiores no Seridó, devido à dominância de sites do tipo riacho nessa região. Pensa-se que os sistemas lóticos são organizados como uma hierarquia aninhada, tanto temporal quanto espacialmente (Johnson et al., 1995; Poff, 1997). Em uma hierarquia aninhada, processos de grande escala, como clima e geomorfologia, definem níveis mais altos de organização dos aspectos físicos e biológicos (por exemplo, morfologia do rio, regime de fluxo, pool de espécies) do ecossistema fluvial (Johnson et al., 1995). Esses níveis mais altos de organização influenciam fatores físicos e biológicos em uma variedade de escalas espaciais e temporais mais baixas, que podem ser particularmente relevantes para sistemas semiáridos (Poff, 1997; Maltchik e Medeiros, 2006). Portanto, em escalas temporais e espaciais relativamente curtas (ou seja, durante as estações secas e úmidas e através de bacias hidrográficas ou trechos de rios), vários fatores podem influenciar as variações no habitat físico disponível para os colonizadores.

No contexto do presente estudo, a Análise de Correspondência Canônica identificou vários níveis de variáveis de captação e alcance de riachos correlacionados com a estrutura do habitat. Ao nível da bacia, as variáveis que representam a ordem do riacho, como comprimento do rio, comprimento do córrego da localização do local e distância do local da fonte, foram importantes determinantes da estrutura do habitat. Na escala de alcance do córrego, a estrutura do habitat foi relacionada com a inclinação da margem e a largura do local, enquanto a profundidade não foi um determinante importante da estrutura do habitat. Como todas as variáveis de habitat foram medidas nas margens, não se deve esperar que a profundidade geral do local seja um elemento importante que afete a estrutura do habitat (a profundidade média foi de $34,4 \pm 19,4$ cm). Entretanto, a inclinação e a largura podem afetar o habitat físico através de sua influência no grau de conectividade entre o habitat aquático e terrestre. É importante lembrar que a declividade foi medida nas margens, compreendendo a porção terrestre das margens. Os parâmetros de qualidade da água em escala local também foram considerados importantes descritores da estrutura do habitat, principalmente aqueles associados ao conteúdo de oxigênio dissolvido na água (nomeadamente, oxigênio dissolvido e velocidade da água). Embora a temperatura e a transparência também possam estar relacionadas ao oxigênio dissolvido, a primeira foi relativamente constante em todos os locais de estudo

(a temperatura média da água foi de $29,5 \pm 2,9$ °C) e a última (estimada a partir das profundidades de Secchi) foi muito alta em todas as áreas de estudo. como com médias por local variando de 16 a 90 cm. O modelo de habitat foi sugerido como uma estrutura onde fatores bióticos influenciam a biota (Southwood, 1977). Em ecossistemas de riachos, a inundação e a variabilidade do fluxo têm sido sugeridas como os principais eixos no modelo de habitat (Minshall, 1988; Poff e Ward, 1989). Embora os resultados apresentados neste estudo indiquem que as variações sazonais associadas ao regime de inundação e fluxo são elementos importantes que afetam a estrutura do habitat, há indícios de que o modelo de habitat pode ser multidimensional. Nos sistemas semiáridos do Brasil, a estrutura do habitat está sendo impulsionada por muitos componentes em diferentes níveis, onde o papel das características de captação e morfologia local e variáveis de água aumentariam em predominância em suas respectivas escalas. Isto realça a importância da ligação entre a estrutura do habitat e a diversidade biótica a nível local e regional.

O presente estudo mostra que a estrutura do habitat em sistemas semiáridos no nordeste do Brasil é composta por uma gama de elementos que variam de acordo com o tipo de habitat e estação do ano, enquanto a heterogeneidade do habitat foi mais forte nas escalas local e de captação do que entre áreas de captação. Portanto, vários níveis da hierarquia de captação são importantes para a estrutura do habitat, desde variáveis de escala de captação, como tamanho e elevação do rio, até características locais, como largura, inclinação e oxigênio dissolvido na água. Tais feições devem ser avaliadas e utilizadas como uma estrutura básica para caracterização e avaliação de habitats aquáticos no semiárido brasileiro. Os resultados apresentados têm implicações para a conservação e manejo dos sistemas semiáridos brasileiros. Dado que o habitat é a estrutura básica para a colonização de organismos aquáticos, os mecanismos potenciais que mantêm a diversidade biótica estão em todos os níveis da bacia hidrográfica. É fundamental, portanto, identificar as partes dos ecossistemas ribeirinhos que são vitais para a manutenção de sua saúde.

Referências

Part II

ESTATÍSTICAS DESCRITIVAS

Chapter 6

R Modulo 1 - Importando planilhas de dados

RESUMO

A importação de planilhas do Excel para o ambiente de programação R é uma tarefa fundamental para análise de dados e estatísticas. Através da importação de planilhas do Excel, é possível transformar dados armazenados em formatos familiares em estruturas que podem ser manipuladas e exploradas de maneira eficaz no R. Isso permite a aplicação de diversas técnicas estatísticas e criação de visualizações informativas, contribuindo para a tomada de decisões embasadas em dados. Neste contexto, entender como importar dados do Excel para o R é um passo crucial para realizar análises de alta qualidade e obter insights significativos a partir dos conjuntos de dados disponíveis.

Apresentação

A importação de planilhas do Excel para o ambiente de programação R é uma tarefa fundamental para análise de dados e estatísticas. O R é uma linguagem de programação amplamente utilizada por cientistas de dados, pesquisadores e analistas para manipular, visualizar e modelar informações. Através da importação de planilhas do Excel, é possível transformar dados armazenados em formatos familiares em estruturas que podem ser manipuladas e exploradas de maneira eficaz no R. Isso permite a aplicação de diversas técnicas estatísticas e criação de visualizações informativas, contribuindo para a tomada de decisões embasadas em dados. Neste contexto, entender como importar dados do Excel para o R é um passo crucial para realizar análises de alta qualidade e obter insights significativos a partir dos conjuntos de dados disponíveis.

6.1 Sobre os dados do PPBio

Usaremos para esse tutorial dados coletados no Programa de Pesquisa em Biodiversidade - PPBio (veja Programa de Pesquisa em Biodiversidade – PPBio). Nesta base de dados estão armazenadas informações sobre diversos grupos taxonômicos distribuídos em diversas unidades amostrais (UA's ou sítios), como peixes, macroinvertebrados bentônicos, quironomídeos e zooplânton, além de dados do habitat, como variáveis físicas e químicas, morfologia do habitat, composição do substrato, estrutura de habitat marginal, entre outros (Figura 6.1).

Essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados para os valores de Captura Por Unidade de Esforço (CPUE), nem foram relativizados ou transformados.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	cited	sp-devs	sp-bimac	sp-hed	ch-bimac	cl-socfa	cl-orient	co-macrop	co-chetar	cr-menaz	cu-hepat	cy-alibe	ge-brasil	he-marg	ho-melad	hy-pular	le-mear	le-plau
2	S-A-ZAI	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	S-A-CCL	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	S-R-CT1	0	194	55	0	0	5	0	1	14	0	0	0	3	0	1	9	0
5	S-R-CT2	0	19	0	0	0	0	0	0	0	0	0	0	0	0	5	0	3
6	S-A-TAI	0	23	1	13	0	0	0	0	0	0	0	0	0	0	0	0	0
7	S-R-CT3	0	142	3	3	0	69	0	0	4	0	0	0	1	17	43	0	1
8	S-R-CP2	0	5	1	0	40	9	0	0	0	0	0	0	0	10	2	0	3
9	S-R-CT4	0	46	0	0	178	0	0	0	0	0	0	0	0	2	2	0	0
10	S-R-CT5	0	206	66	0	0	25	0	0	0	0	0	0	0	31	11	0	2
11	S-R-CP3	0	16	0	0	13	24	0	0	0	0	0	0	0	4	0	0	1
12	S-A-TA3	0	234	7	238	0	0	2	0	0	0	0	0	0	20	0	0	0
13	S-R-CT4	0	0	1	0	0	5	0	1	0	0	50	3	1	4	3	0	0
14	S-R-CT5	0	0	0	0	14	6	0	0	0	0	0	0	0	2	0	0	2
15	S-A-TA4	0	394	0	273	0	0	0	0	1	0	0	0	1	9	0	0	2
16	S-B-MU1	0	12	0	0	0	0	0	0	0	0	0	0	190	0	0	0	0
17	S-B-ET1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	S-B-GU1	0	2	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0
19	S-B-PC2	0	44	0	0	2	0	0	0	0	0	0	0	8	0	0	2	0
20	S-B-MU2	0	99	0	0	0	0	0	0	0	0	0	67	0	1	0	0	0
21	S-B-GU2	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0
22	S-B-PC3	0	75	7	0	0	4	0	0	0	41	0	16	0	2	0	0	0
23	S-A-MU3	0	511	0	0	0	0	0	0	0	0	0	145	0	0	0	0	0
24	S-B-GU3	0	6	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0
25	S-B-PC4	0	7	17	0	0	0	0	0	0	0	0	81	5	0	1	0	1
26	S-A-MU4	0	235	0	0	0	0	0	0	0	0	0	509	0	0	0	0	0
27	S-A-GU4	0	13	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0

Figure 6.1: Parte da planilha de dados brutos do PPBio.



A planilha `ppbio*-peixes.***` contém o delineamento amostral de um dos estudos do Projeto PPBio (Figura 6.2). Nas linhas são apresentadas as abreviações dos nomes das unidades amostrais (UA's) e nas colunas são apresentados os nomes abreviados das espécies - temos portanto uma matriz comunitária. No corpo da planilha temos os valores para o tipo de dados amostrado. Quantitativo, semi-quantitativo ou qualitativo.

Qual desses tipos de dados você acha que é apresentado na planilha?

Várias das espécies nessa matriz tem grande importância ecológica, como é o caso de *Astyanax bimaculatus*¹ (Figura 6.3), que é muito comum em rios intermitentes e serve de alimento para predadores maiores como a espécie *Hoplias malabaricus*² (Figura 6.4).

¹ A etimologia do gênero *Astyanax* vem da mitologia Grega. Heitor personagem da “Ilíada”,

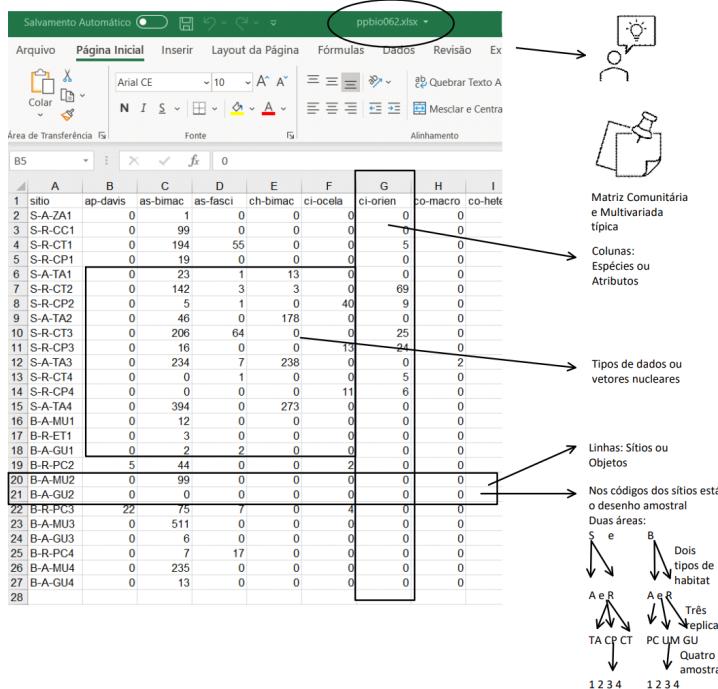


Figure 6.2: Associação entre a planilha de dados brutos do PPBio e o delineamento amostral do estudo.



Figure 6.3: **Astyanax bimaculatus**, a espécie mais comum da matriz de dados ppbio. Peru, by Eakins, R. Fonte: <<https://www.fishbase.se/summary/Astianax-bimaculatus.html>>



Figure 6.4: **Hoplias malabaricus**, espécie que cresce para se tornar um importante predador. Brazil, by Roselet, F.F.G. Fonte: <<https://www.fishbase.se/summary/Hoplias-malabaricus.html>>



NOTA

Se você ainda não tem o R e o RStudio instalados veja a seção Instalação do R e RStudio do Capítulo Introdução ao R/RStudio

6.2 Importando a planilha de trabalho



ATENÇÃO

Os links para baixar as planilhas necessárias para repetir esse tutorial podem ser encontrados na seção Arquivos disponíveis do Capítulo Bases de dados.

Ou, baixe aqui o arquivo ppbio06c-peixes.xlsx

Para começar a usar o R e analisar os dados do Projeto PPBio, abra o RStudio,

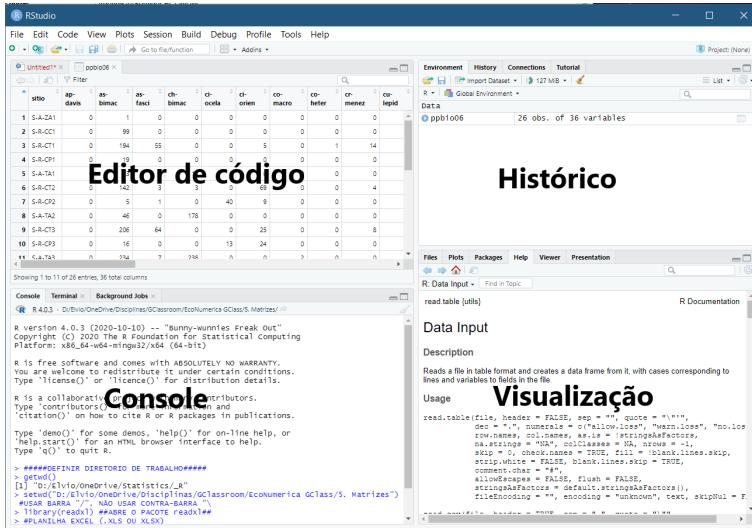


Figure 6.5: Interface típica do RStudio e nome dos painéis ou janelas.

verifique sua interface (Figura 6.5) e siga as instruções a seguir.

6.3 Mensagens de erro e avisos no R

No contexto da linguagem de programação R, mensagens de erro (errors) e mensagens de aviso (warnings) que aparecem em vermelho no painel de console. Elas são formas de feedback do sistema que indicam problemas ou situações potencialmente problemáticas durante a execução do código. Aqui está uma breve explicação de cada um:

1. Erro (Error):

- Um erro ocorre quando algo no código não está correto ou não pode ser executado como esperado.
- Isso pode ser causado por sintaxe incorreta, uso incorreto de funções, operações inválidas, referências a objetos que não existem, entre outros problemas.
- Quando ocorre um erro, a execução do código é interrompida e uma mensagem de erro é exibida no console em vermelho, indicando o tipo de erro e, muitas vezes, a linha onde ocorreu.

2. Aviso (Warning):

- Não indica erro. Um aviso é emitido quando algo no código pode resultar em um comportamento indesejado ou em resultados inesperados, mas não interrompe necessariamente a execução do código.

tinha um filho chamado Astianax.

²Do Grego, *hoplon*, arma ou armadura, em referência aos dentes caniniformes muito desenvolvidos, e forte estrutura óssea na cabeça.

- Os avisos geralmente indicam situações que merecem atenção, como conversões de tipos de dados que podem perder informações ou funções que estão sendo usadas de maneira que pode levar a resultados questionáveis.
- Os avisos são exibidos em vermelho no console e fornecem informações sobre a natureza do aviso e, possivelmente, como abordá-lo.

É importante prestar atenção a mensagens de erro e avisos, pois eles fornecem insights sobre problemas em seu código ou potenciais fontes de comportamento inesperado. Resolver erros é fundamental para que o código funcione conforme o esperado. Embora os avisos não interrompam a execução, investigá-los pode ajudar a evitar problemas futuros ou melhorar a qualidade do código.

6.4 Organização básica

No ambiente do RStudio no painel de edição de código execute (**Ctrl+Enter** com o teclado ou Run no editor de código) os comandos a seguir, para instalar os pacotes necessários para este módulo.

```
install.packages("readxl") #importa arquivos do excel
```

E em seguida,

```
library(readxl)
```

Os códigos acima, são usados para instalar e carregar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades do R (R CRAN) (TEAM, 2017), e RStudio (TEAM, 2022). No exemplo acima, o pacote `readxl` permite ler e escrever arquivos Excel no R.

Para instalar um pacote no R, você precisa usar a função `install.packages()`. Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`.

Por exemplo, para carregar o pacote `readxl`, você precisa executar a função `library(readxl)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

Usaremos uma matriz multivariada (sítios x espécies, matriz comunitária) do Projeto PPBio chamada ppbio**.xlsx que está no diretório “C:/Meu/Diretório/De/Trabalho/Planilha.xlsx”

Note que o símbolo # em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações.

Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

Definindo o diretório de trabalho e installando os pacotes necessários:

```
getwd()  
setwd("C:/Seu/Diretório/De/Trabalho")
```

Alternativamente você pode ir na barra de tarefas e escolhes as opções:\SESSION -> SET WORKING DIRECTORY -> CHOOSE DIRECTORY

6.4.1 Prefira sempre códigos e scripts do que mouse e menus de janelas no R

Optar pelo uso de scripts e comandos de teclado no R, em vez das opções baseadas em mouse e menus das janelas, oferece várias vantagens significativas para quem está envolvido em análises de dados e programação. Aqui estão algumas justificativas para essa abordagem:

1. **Reprodutibilidade:** O uso de scripts permite que todas as etapas de análise e manipulação de dados sejam documentadas em um único lugar. Isso facilita a reexecução de todo o processo, tornando a análise reproduzível e permitindo que outras pessoas compreendam e validem o trabalho realizado.
2. **Automação:** Comandos de script podem ser facilmente repetidos ou adaptados para diferentes conjuntos de dados. Isso possibilita a automação de tarefas complexas, economizando tempo e reduzindo a possibilidade de erros manuais.
3. **Flexibilidade:** Enquanto as opções de mouse e menus podem ser limitadas em termos das ações específicas que permitem, os scripts oferecem uma flexibilidade muito maior. Você pode personalizar cada etapa do processo de análise de acordo com suas necessidades específicas.
4. **Eficiência:** A digitação de comandos é geralmente mais rápida do que navegar por menus e clicar em botões, especialmente quando se trata de tarefas repetitivas e/ou complexas.
5. **Controle total:** Ao utilizar scripts, você tem controle total sobre cada etapa do processo. Isso é particularmente importante em análises estatísticas, onde pequenas variações nos parâmetros podem ter um grande impacto nos resultados.

6. **Aprendizado contínuo:** Escrever e modificar scripts permite um maior aprendizado e domínio da linguagem R. Conforme você ganha experiência, poderá realizar análises mais sofisticadas e explorar recursos avançados.
7. **Portabilidade:** Scripts podem ser facilmente compartilhados com outros pesquisadores ou colegas, independentemente do sistema operacional utilizado. Isso torna a colaboração mais fluida e ajuda a evitar problemas de compatibilidade.
8. **Melhor entendimento:** Ao escrever e ler scripts, você desenvolve uma compreensão mais profunda dos processos que está realizando. Isso é importante para identificar possíveis erros e interpretar corretamente os resultados.
9. **Documentação clara:** Ao escrever um script, você pode adicionar comentários explicativos que descrevem cada passo e sua lógica. Isso resulta em uma documentação clara e autoexplicativa do trabalho realizado.
10. **Consistência:** O uso de scripts promove a adoção de práticas consistentes em toda a análise, reduzindo a chance de erros causados por abordagens diferentes em momentos distintos.

Em resumo, a abordagem baseada em scripts e comandos de teclado oferece mais controle, flexibilidade, eficiência e reproduzibilidade, tornando-a a escolha preferida para profissionais que buscam análises de dados precisas, consistentes e de alta qualidade no ambiente R.

6.5 Importando a planilha

```
library(readxl)
ppbio06 <- read_excel("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrices/ppbio06c-"
                       sheet = "Sheet1",
                       na = "NA")
str(ppbio06)
class(ppbio06)
```

Com essas linhas de código a primeira coluna da matriz importada apresenta texto. Não queremos assim porque vamos fazer cálculos matemáticos na matriz.

Resolvemos o problema com mais algumas linhas de código.

```
ppbio06 <- as.data.frame(ppbio06)
class(ppbio06)
rownames(ppbio06) <- ppbio06[,1] #tem que ser um df
ppbio06[,1] <- NULL
```

Ou podemos instalar esse pacote de importação de arquivos .xlsx para o R.

```

install.packages("openxlsx")

library(openxlsx)

## Warning: package 'openxlsx' was built under R version 4.3.2

ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06c-peixes.xlsx",
                    rowNames = T,
                    colNames = T,
                    sheet = "Sheet1")

str(ppbio)
class(ppbio)
ppbio_ma <- as.matrix(ppbio) #lê ppbio como uma matriz
str(ppbio_ma)
class(ppbio_ma)
#ppbio
#ppbio_ma

```

Compare as diferenças. Agora podemos exportar os dados como uma matriz de dados em formato de valores separados por vírgula (.csv).

```

write.table(ppbio, "ppbiocsv.txt", append = F, quote = T, ";", row.names = T)
dir <- getwd()
shell.exec(dir) #abre o diretório de trabalho no Windows Explorer

```

Podemos abrir o arquivo .csv criado ppbiocsv.txt usando os códigos abaixo.

```

ppbiocsv <- read.csv("ppbiocsv.txt",
                      sep = ";", dec = ",", #definimos o dígito separador
                      header = T,
                      row.names = 1,
                      na.strings = NA)

str(ppbiocsv)
ppbiocsv

```

Lembre de prestar atenção no dígito separador de decimais ”, ” ou ”. ”. Além disso, só estaamos usando ppbio***.*** porque o diretório de trabalho ja fo definido no início. Se não deveríamos estar usando C:/Seu/Diretório/De/Trabalho/ppbio***.***

Alguns comandos para exibir a planilha são “case-sensitive” (ignore.case(object))

```
#View(ppbio)
print(ppbio)
ppbio
str(ppbio)
##View
##view
##remove
```

6.5.1 Outra forma de achar e importar uma planilha

Essa forma é desaconselhável porque é demorada e sujeita a erros. Além de precisar ser refeita sempre que se quiser abrir uma nova planilha ou reabrir a última planilha importada. Veja o tópico Prefira sempre códigos e scripts do que mouse e menus de janelas no R

```
getwd()
ppbio <- read.xlsx(file.choose(), #abre o windows explorer
                   rowNames = T, colNames = T,
                   sheet = "Sheet1")
```

6.6 Importando .ods do LibreOffice Calc

A planilha a seguir pode ser baixada da seção ??

```
#install.packages("readODS")

library(readODS)
ppbio06.ods <- read_ods("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizess/ppbio06.ods")
               row.names = TRUE,
               col.names = TRUE,
               sheet = "Sheet1",
               as_tibble = FALSE,
               na = "n/a") # porque existem celulas vazias (n/a)
ppbio06.ods <- na.omit(ppbio06.ods)
str(ppbio06.ods)
class(ppbio06.ods)
ppbio06.ods_ma <- as.matrix(ppbio06.ods) #lê como uma matriz
str(ppbio06.ods_ma)
class(ppbio06.ods_ma)
#ppbio06.ods
#ppbio06.ods_ma
```

6.6.1 Colunas com o mesmo nome

Em algumas situações é necessário encontrar colunas com o mesmo nome em um conjunto de dados e soma-las ou fazer sua média. Isso pode ocorrer quando se está trabalhando com dados de várias fontes e é necessário combinar esses diferentes conjuntos de dados.

Considere por exemplo um cenário onde se está trabalhando em um projeto de análise de dados ecológicos e você recebeu conjuntos de dados de diferentes locais de amostragens enviados por diferentes pesquisadores, e cada pesquisador enviou seu conjunto de dados que contém informações sobre as mesmas espécies (ou variáveis ambientais). Devido a diferentes sistemas de registro ou a falta de comunicação entre os pesquisadores, pode haver repetição nos nomes das colunas.

```
#Dados de mamíferos roedores
df <- data.frame(
  Rato = c(1, 2, 3),
  Musaranho = c(4, 5, 6),
  Rato = c(7, 8, 9), #nome duplicado
  Esquilo = c(0, 0, 1),
  Ratão = c(1, 1, 0),
  Castor = c(1, 0, 0),
  Tâmia = c(11, 12, 13),
  Marmota = c(1, 2, 0),
  Castor = c(2, 1, 1), #nome duplicado
  check.names = FALSE
)
df
```

	Rato	Musaranho	Rato	Esquilo	Ratão	Castor	Tâmia	Marmota	Castor	
## 1	1		4	7	0	1	1	11	1	2
## 2	2		5	8	0	1	0	12	2	1
## 3	3		6	9	1	0	0	13	0	1

Por exemplo, no estudo sobre mamíferos roedores acima, quando se combina o conjunto geral de dados para realizar uma análise abrangente, depara-se com colunas duplicadas, onde a matriz com o conjunto total de dados contém espécies repetidas.

Nesse caso, encontrar e resolver colunas com o mesmo nome é crucial para garantir a integridade dos dados e realizar uma análise precisa. Você deve consolidar essas colunas duplicadas, somando-as ou fazendo sua média.

```
# Achando colunas com nomes duplicados
dup_cols <- names(df)[duplicated(names(df))]
# Somando colunas com o mesmo nome
```

```

for (col_name in unique(dup_cols)) {
  # Get indices of columns with the same name
  col_indices <- which(names(df) == col_name)
  # Sum columns with the same name
  df[[col_name]] <- rowSums(df[, col_indices], drop = FALSE)
}
# Remove as colunas duplicadas originais e mantem as novas colunas que são a soma ("ex"
df <- df[, !duplicated(names(df))]
# Mostra a nova tabela com colunas repetidas somadas
print(df)

##   Rato Musaranho Esquilo Ratão Castor Tâmia Marmota
## 1     8          4      0     1     3    11      1
## 2    10          5      0     1     1    12      2
## 3    12          6      1     0     1    13      0

```

6.6.2 Removendo linhas ou colunas por nome

```

#Colunas
df <- subset(df, select = -Lec.mono) #escolhendo uma coluna pelo nome

#Linhas
del_rows <- c("ST20")
del_rows
m_part <- df[!(row.names(entorno) %in% c(del_rows)),]
m_part

```

6.6.3 Criando uma matriz de médias

Por razões diferentes o procedimento anterior pode vir a ser necessário de ser aplicado às linhas. Por exemplo, quando se quer somar ou fazer a média de amostras diferentes do mesmo ambiente de coleta. Veja a matriz abaixo.

```

data <- read.table(text =
  Sp1 Sp2 Sp3 Sp4 Sp5 Sp6 Sp7 Sp8
  A1 0 0 0 0 0 6 1
  A2 0 0 0 2 0 0 10 2
  B1 93 2 0 177 0 260 2 5
  B2 0 4 0 8 0 0 83 7
  C1 0 0 0 0 1 0 0 1
  C2 0 0 1 0 0 1 0 1
  C3 0 2 0 2 0 0 0 1
", header = TRUE, row.names = 1)

```

```
data
```

```
##      Sp1 Sp2 Sp3 Sp4 Sp5 Sp6 Sp7 Sp8
## A1    0   0   0   0   0   0   6   1
## A2    0   0   0   2   0   0  10   2
## B1   93   2   0 177   0 260   2   5
## B2    0   4   0   8   0   0  83   7
## C1    0   0   0   0   1   0   0   1
## C2    0   0   1   0   0   1   0   1
## C3    0   2   0   2   0   0   0   1

library("tidyverse")
#Inserindo coluna para agrupamentos
nrow(data); ncol(data) #no. de N colunas x M linhas
data_g <- cbind(Grupos = rownames(data), data)
data_g

grps <- substr(data_g[, 1], 1,3)
grps

data_g <- data_g %>% mutate(Grupos=c(grps))

#data_avg <- aggregate(data_g[, 9:9], list(data_g$Grupos), mean)
#data_avg

data_avg <- data_g %>%
  group_by(Grupos) %>%
  summarise(across(.cols = everything(), ~ mean(.x, na.rm = TRUE)))
data_avg

data_dp <- data_g %>%
  group_by(Grupos) %>%
  summarise(across(.cols = everything(), list(mean = mean, sd = sd)))
#?across
data_dp

#Primeira coluna para nomes das linhas
data_dp <- as.data.frame(data_dp)
class(data_dp)
rownames(data_dp) <- data_dp[,1]
data_dp[,1] <- NULL
data_dp <- round(data_dp, 1)
data_dp
#Salvando a matriz
```

```

write.table(data_dp,
            "data_dp.csv",
            append = F,
            quote = TRUE,
            sep = ";", dec = ",",
            row.names = T)
data_dp_csv <- read.csv("data_dp.csv",
                        sep = ";", dec = ",",
                        header = T,
                        row.names = 1,
                        na.strings = NA)

```

Apêndices

Sites consultados

[<https://youtu.be/U6ksXvvY6Q0>]
[\[https://youtu.be/a7EJE_2mtGk\]](https://youtu.be/a7EJE_2mtGk)

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executa-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

```

## install.packages("readxl") #importa arquivos do excel
## library(readxl)
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
library(readxl)
ppbio06 <- read_excel("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06.xlsx")
                                         sheet = "Sheet1",
                                         na = "NA")
str(ppbio06)
class(ppbio06)
ppbio06 <- as.data.frame(ppbio06)
class(ppbio06)
rownames(ppbio06) <- ppbio06[,1] #tem que ser um df
ppbio06[,1] <- NULL
#install.packages("openxlsx")
library(openxlsx)
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/bentos06.xlsx",
                   rowNames = T,

```

```
    colNames = T,
    sheet = "contagem")
str(ppbio)
class(ppbio)
ppbio_ma <- as.matrix(ppbio) #lê ppbio como uma matriz
str(ppbio_ma)
class(ppbio_ma)
#ppbio
#ppbio_ma
## write.table(ppbio, "ppbiocsv.txt", append = F, quote = T, ";", row.names = T)
## dir <- getwd()
## shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
## ppbiocsv <- read.csv("ppbiocsv.txt",
##                      sep = ";", dec = ",", #definimos o dígito separador
##                      header = T,
##                      row.names = 1,
##                      na.strings = NA)
## str(ppbiocsv)
## ppbiocsv
## #View(ppbio)
## print(ppbio)
## ppbio
## str(ppbio)
## #?View
## #?view
## #?remove
## getwd()
## ppbio <- read.xlsx(file.choose(), #abre o windows explorer
##                     rowNames = T, colNames = T,
##                     sheet = "Sheet1")
```

Referências

Chapter 7

R Modulo 2 - Tipos de dados no R

Apresentação

O R é uma **linguagem de expressão** com uma sintaxe muito simples (veja TEAM (1999-2024)). Ela é sensível a maiúsculas e minúsculas, como a maioria dos pacotes baseados em UNIX, então A e a são símbolos diferentes e se refeririam a variáveis diferentes. O conjunto de símbolos que podem ser usados nos nomes do R depende do sistema operacional e do país em que o R está sendo executado (teoricamente, da localidade em uso). Normalmente, todos os símbolos alfanuméricos são permitidos (e em alguns países isso inclui letras acentuadas), além de . e _, com a restrição de que um nome deve começar com . ou uma letra, e se começar com ., o segundo caractere não deve ser um dígito. Os nomes são efetivamente ilimitados em comprimento.

ATENÇÃO Evite usar nomes longos e complicados

Comandos elementares consistem em expressões ou atribuições. Se uma **expressão** for dada como um comando, ela é avaliada, impressa (a menos que seja especificamente tornada invisível) e o valor é perdido. Uma **atribuição** também avalia uma expressão e passa o valor para uma variável, mas o resultado não é automaticamente impresso.

Os comandos são separados por um ponto e vírgula (;) ou por uma nova linha (**Enter**). Comandos elementares podem ser agrupados em uma única expressão composta por chaves ({' and '}).

Comentários podem ser colocados em quase qualquer lugar, começando com uma marca de hash (#), e tudo até o final da linha é um comentário.

Se um comando não estiver completo no final de uma linha, o R fornecerá um prompt diferente, por padrão + (ao invés do tradicional >, nas linhas seguintes

e continuará a ler a entrada até que o comando seja sintaticamente completo. Este prompt pode ser alterado pelo usuário. Geralmente, omitiremos o prompt de continuação e indicaremos a continuação apenas recuando.

As linhas de comando digitadas no console são limitadas a cerca de 4095 bytes (não caracteres).

No R, uma linguagem de programação amplamente utilizada para análise de dados e estatísticas, os tipos e estruturas de dados desempenham um papel fundamental na manipulação e organização das informações.

7.1 Tipos e estruturas de dados no R

No R, uma linguagem de programação amplamente utilizada para análise de dados e estatísticas, os tipos e estruturas de dados desempenham um papel fundamental na manipulação e organização das informações. Você vai conhecer esses tipos de estruturas de dados a seguir, mas antes se familiarize com como o R funciona.

Para começar a usar o R e analisar os dados do Projeto PPBio, abra o RStudio, verifique sua interface (Figura 7.1) e siga as instruções a seguir.

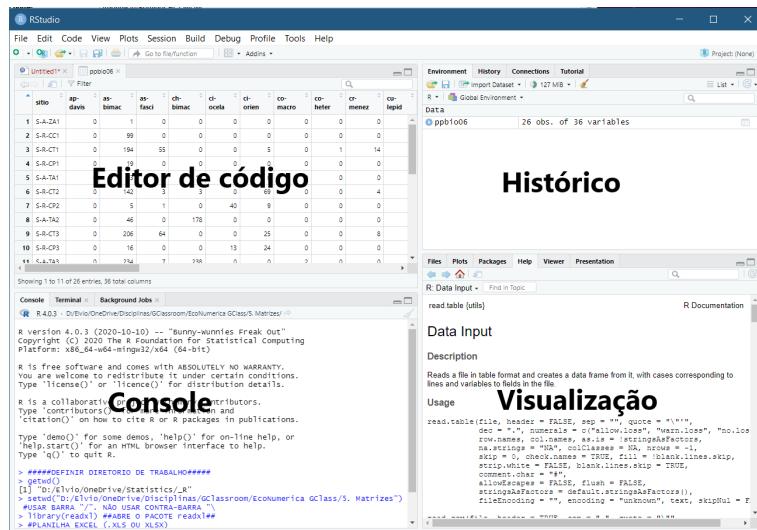


Figure 7.1: Interface típica do RStudio e nome dos painéis ou janelas.

7.2 Mensagens de erro e avisos no R

No contexto da linguagem de programação R, mensagens de erro (errors) e mensagens de aviso (warnings) que aparecem em vermelho no painel de console.

7.3. O QUE É UM SCRIPT E QUAL SUA DIFERENÇA PARA UM CÓDIGO?95

Elas são formas de feedback do sistema que indicam problemas ou situações potencialmente problemáticas durante a execução do código. Aqui está uma breve explicação de cada um:

1. Erro (Error):

- Um erro ocorre quando algo no código não está correto ou não pode ser executado como esperado.
- Isso pode ser causado por sintaxe incorreta, uso incorreto de funções, operações inválidas, referências a objetos que não existem, entre outros problemas.
- Quando ocorre um erro, a execução do código é interrompida e uma mensagem de erro é exibida no console em vermelho, indicando o tipo de erro e, muitas vezes, a linha onde ocorreu.

2. Aviso (Warning):

- Não indica erro. Um aviso é emitido quando algo no código pode resultar em um comportamento indesejado ou em resultados inesperados, mas não interrompe necessariamente a execução do código.
- Os avisos geralmente indicam situações que merecem atenção, como conversões de tipos de dados que podem perder informações ou funções que estão sendo usadas de maneira que pode levar a resultados questionáveis.
- Os avisos são exibidos em vermelho no console e fornecem informações sobre a natureza do aviso e, possivelmente, como abordá-lo.

É importante prestar atenção a mensagens de erro e avisos, pois eles fornecem insights sobre problemas em seu código ou potenciais fontes de comportamento inesperado. Resolver erros é fundamental para que o código funcione conforme o esperado. Embora os avisos não interrompam a execução, investigá-los pode ajudar a evitar problemas futuros ou melhorar a qualidade do código.

7.3 O que é um script e qual sua diferença para um código?

Em programação R, tanto “código” quanto “script” se referem a sequências de instruções escritas na linguagem de programação R (veja também Snippet e chunk). No entanto, há uma diferença sutil em como esses termos são geralmente usados:

1. Código:

Em R, “código” geralmente se refere a linhas individuais ou blocos de instruções de programação R que realizam tarefas ou operações específicas. O código pode ser escrito de forma interativa em um console R ou dentro de um arquivo de script. O código pode consistir em declarações simples, como atribuir valores a variáveis, realizar cálculos, definir funções ou chamar funções de pacotes.

2. Script:

Um “script” em R refere-se a um arquivo contendo uma série de instruções de código R. Scripts R são essencialmente arquivos de texto contendo uma seqüência de comandos e declarações R que podem ser executados em conjunto. Os scripts R permitem que os usuários organizem seu código em unidades reutilizáveis e estruturadas. Os scripts R geralmente têm a extensão de arquivo “.R”.

Em resumo, enquanto “código” se refere a instruções individuais de programação R, “script” se refere a um arquivo contendo uma coleção de código R, geralmente usado para executar tarefas ou análises maiores de forma estruturada e organizada.

Agora você pode começar a escrever seu código no console. Digite “Início do R Módulo” no console clique em Run (no canto superior direito do painel de edição de código) ou **Ctrl+Enter** e veja o que acontece:

```
"Início do R Módulo"
```

```
## [1] "Início do R Módulo"
```

O R mostra o que você escreveu, e apresenta este [1] do lado. Isso significa que seu resultado foi apresentado na primeira linha. Como você pode ver, o R imprime de volta (no Console) o resultado de comandos que você dá. Você agora pode tentar as operações básicas.

```
600+66
```

```
## [1] 666
```

Cálculos repetitivos podem ser automatizados usando um loop

```
x <- 10
x
```

```
## [1] 10
```

```
for (i in 1:5) {x+1->x}
x
```

```
## [1] 15
```

Aqui o número 1 foi somado cinco vezes ao valor de x que foi armazenado cada vez como x (substituindo o valor anterior) até chegar em 15. Na prática o R calculou $10+1=11$, $11+1=12$, $12+1=13$, $13+1=14$, $14+1=15$.

Note que os : foi usado para produzir uma seqüência de números de 1 a 5.

```
1:5
```

```
## [1] 1 2 3 4 5
```

O R usa os seguintes símbolos para as operações básicas:

```
`+` Adição
`-` Subtração
`/` Divisão
`*` Multiplicação
`^` Potência
`sqrt` Raiz quadrada (função)
`log` Logaritmo (função)
```

Teste agora outras possibilidades com as operações básicas (veja BEASLEY (2004) pags 7-10 para outros exemplos de cálculos básicos).

Você já deve estar percebendo o potencial da programação em R como ferramenta para aprender estatística ou análise de dados, duas áreas com muita demanda de profissionais.

Em sua apostila “*Conversando com o R usando 57 palavras*” MELO (2014) ressalta que a programação em R “ajuda no sentido de prover métodos para resolvemos um problema ou entendermos uma análise”. Isso porque, primeiro, temos que reunir os elementos fundamentais e informações necessários, depois identificar e dar nomes a esses elementos, para apenas depois disso, decompor nosso problema em suas partes e, finalmente, colocar essas partes em sequência lógica (MELO, 2014).

A forma linear (passo-a-passo), mas hierarquicamente estruturada, da programação em R, permite uma construção intuitiva e lógica do raciocínio associado a cada análise, que, com o tempo, se torna parte cotidiana das nossas análises de dados. Assim como, na Língua Portuguesa, aprendemos o alfabeto, depois as palavras e eventualmente conversamos sem precisar pensar sobre a gramática e construção das frases.

7.4 Tipos e estruturas de dados no R

O R oferece uma variedade de tipos e estruturas de dados que permitem aos cientistas de dados e analistas manipular e organizar informações de forma eficiente. A escolha da estrutura de dados correta depende das necessidades específicas de análise e do tipo de dados que você está lidando.

1. Tipos de Dados Básicos:

- **Numéricos:** Representam valores numéricos, como inteiros ou números de ponto flutuante.
- **Caracteres:** Armazenam texto e são usados para representar sequências de caracteres.
- **Lógicos:** Armazenam valores lógicos (TRUE ou FALSE), úteis para expressar afirmações condicionais.

- **Complexos:** Armazenam números complexos.
- 2. Vetores:**
- Os vetores são a estrutura de dados mais básica no R.
 - Eles podem conter elementos de um único tipo de dado (exemplo: vetor numérico, de caracteres ou lógico).
 - Os vetores são criados usando a função `c()`.
- 3. Matrizes:**
- Matrizes são vetores bidimensionais, onde os elementos são organizados em linhas e colunas.
 - Todos os elementos de uma matriz devem ser do mesmo tipo de dado.
 - Matrizes podem ser criadas usando a função `matrix()`.
- 4. Data Frames:**
- Data frames são estruturas de dados tabulares semelhantes a planilhas ou tabelas de banco de dados.
 - Cada coluna de um data frame pode conter um tipo de dado diferente.
 - Data frames são amplamente usados para armazenar e manipular conjuntos de dados.
 - São frequentemente criados com funções como `data.frame()` ou lidos de arquivos externos.
- 5. Fatores:**
- Fatores são usados para representar variáveis categóricas ou de fatores.
 - São úteis para análises estatísticas e gráficos.
 - Podem ser criados com a função `factor()`.
- 6. Listas:**
- Listas são estruturas de dados flexíveis que podem conter elementos de diferentes tipos.
 - Os elementos de uma lista podem ser acessados por meio de índices ou nomes.
 - Listas são criadas usando a função `list()`.
- 7. Arrays:**
- Arrays são estruturas multidimensionais que podem conter elementos de um único tipo de dado.
 - São semelhantes às matrizes, mas podem ter mais de duas dimensões.
 - Arrays são criados com a função `array()`.
- 8. Tabelas Hash:**
- Tabelas hash são estruturas de dados que associam chaves a valores.
 - Não são nativas do R, mas podem ser implementadas com pacotes como o `hash`.

7.5 Tipos de dados no R

7.5.1 Variáveis e tipos básicos de dados (Figura 7.2)

1. **Tipos de Dados Básicos:**

- **Numéricos:** Representam valores numéricos, como inteiros ou números de ponto flutuante.
- **Caracteres:** Armazenam texto e são usados para representar sequências de caracteres.
- **Lógicos:** Armazenam valores lógicos (TRUE ou FALSE), úteis para expressar afirmações condicionais.
- **Complexos:** Armazenam números complexos.

Para guardar os valores em uma variável em R, usamos `<-` (operador de atribuição). Neste caso, o print automático não ocorre, e caso queira checar qual é o valor daquela variável, basta digitar seu nome novamente, ou utilizar o comando `print(nome_da_variavel)`.

Vejamos:

```
a <- 5 + 3
a
print(a)
class(a)
```

```
## [1] 8
## [1] 8
## [1] "numeric"
```

Criamos um **objeto** do R chamado `a` que armazena o resultado de `5+3`. Objetos são palavras ou letras às quais são atribuídos dados. A atribuição possibilita a manipulação de dados ou armazenamento dos resultados de análises (SILVA et al., 2022).

O objeto `a` é uma variável numérica, como você pode ver ao usar a função `class()`. Se quiser trabalhar com inteiros (`integer`), você deve defini-los como `integer`, através da função `as.integer()`.

Vejamos:

```
var1 <- 3
var2 <- as.integer(3)
class(var1)
class(var2)
```

```
## [1] "numeric"
## [1] "integer"
```

A função `class()` tem o mesmo efeito que a função `mode()`. Faça o teste.

Em R, os dados também podem ser do tipo `character`, que é uma variável de texto (em outras linguagens pode ser conhecida como `string`) e `logical` (que em outras linguagens pode ser conhecida como `boolean`), e que assume os valores `TRUE` ou `FALSE`.

Exemplos:

```
b <- "Ola mundo"
c <- TRUE
class(b)
class(c)
```

```
## [1] "character"
## [1] "logical"
```

Como você já deve ter percebido, uma grande parte da sua interação com o R é feita por comandos ou **funções**, onde cada função realiza um determinado papel. Seja comandar o R a mostrar uma variável recem criada, função **print()**, seja para informar o resultado de uma operação, função **log()**.

Uma função vem sempre seguida de parênteses () (no formato, **função(x)**), dentro dos quais colocamos os **argumentos** da função. Os argumentos, por sua vês, devem ser separados por vírgula ,. Os argumentos são informações adicionais que especificam certas particularidades para serem executadas pelas funções.

Usando a função **log()** como exemplo, temos:

```
log(x=20, base=10)
```

```
## [1] 1.30103
```

O argumento **x=** indica de qual valor queremos o log e o argumento **base=** indica qual base para o logaritmo o R vai usar.

Ou, em um segundo caso, podemos ter simplesmente a função sem nenhuma indicação de argumento:

```
log(20)
```

```
## [1] 2.995732
```

Muitas funções possuem opções padrões (*default*) para alguns de seus argumentos. No caso do log, se não for dado nenhum argumento o **default** do R é usar o log natural do valor entre parêntesis.

7.6 Estruturas de dados no R

Agora vamos falar das estruturas de dados no R (Figura 7.3).

2. Vetores:

- Os vetores são a estrutura de dados mais básica no R.

- Eles podem conter elementos de um único tipo de dado (exemplo: numérico, caracteres ou lógico).
- Os vetores são criados usando a função `c()`.

Como já foi dito, os vetores são uma sequência simples de elementos do mesmo tipo. Quando definimos uma variável como fizemos anteriormente, é criado um vetor com um elemento (vetor numérico, vetor de caractere, etc.).

O R opera em **estruturas de dados nomeadas**. A estrutura mais simples é o **vetor numérico**, que é uma entidade única consistindo de uma coleção ordenada de números. Para configurar um vetor chamado `x`, por exemplo, consistindo de cinco números, a saber, 1.4, 6.5, 3, 4 e 71.2, usa-se o comando R:

```
x <- c(1.4, 6.5, 3, 4, 71.2)
```

Esta é uma instrução de atribuição usando a função `c()` (“combinar”), que neste contexto pode receber um número arbitrário de argumentos de vetor e cujo valor é um vetor obtido pela concatenação de seus argumentos de ponta a ponta.

Observe que o operador de atribuição (`<-`), que consiste nos dois caracteres ‘<’ (“menor que”) e ‘-’ (“menos”) ocorrendo estritamente lado a lado, e ele “aponta” para o objeto que recebe o valor da expressão. Na maioria dos contextos, o operador ‘`=`’ pode ser usado como uma alternativa.

Observe:

```
var1 <- 3
is.vector(var1)
```

```
## [1] TRUE
```

Isso explica aquele `[1]` que vimos no início do módulo. Quando você associa uma variável dessa forma, o R cria um vetor, e o valor que você definiu será o primeiro item deste vetor.

Para criarmos vetores com mais de um elemento, incluiremos os valores desejados dentro de um `c()` que significa **combinar**.

Veja o exemplo:

```
var1 <- c(3,6, 7.8, 332)
print(var1)
var2 <- c("Ola", "tudo", "bem")
print(var2)
var3 <- c(TRUE, FALSE, TRUE, TRUE, FALSE, FALSE, TRUE, TRUE, TRUE)
print(var3)

## [1] 3.0 6.0 7.8 332.0
## [1] "Ola" "tudo" "bem"
```

```
## [1] TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE
```

Caso você tente misturar os tipos, o R irá forçar para que os elementos sejam todos do mesmo tipo. Veja alguns exemplos:

3. Matrizes:

- Matrizes são vetores bidimensionais, onde os elementos são organizados em linhas e colunas.
- Todos os elementos de uma matriz devem ser do mesmo tipo de dado.
- Matrizes podem ser criadas usando a função `matrix()`.

As matrizes são estruturas que correspondem às matrizes matemáticas, conjuntos de elementos com linhas e colunas. Assim como os vetores, todos os seus elementos são do mesmo tipo. Existem algumas formas de criar uma matriz:

```
mat1 <- matrix(c(1,5,10,30,15,8),
  nrow=3,
  ncol=2,
  byrow=TRUE)
print(mat1)
```

```
##      [,1] [,2]
## [1,]     1     5
## [2,]    10    30
## [3,]    15     8
```

Ou

```
vec1 <- c(1, 5)
vec2 <- c(10, 30)
vec3 <- c(15, 8)
mat2 <- rbind(vec1, vec2, vec3)
print(mat2)
class(mat2)
```

```
##      [,1] [,2]
## vec1     1     5
## vec2    10    30
## vec3    15     8
## [1] "matrix" "array"
```

Ou

```
a <- sample(c(0,1,5),10,replace=TRUE)
b <- sample(c(0,10,20),10,replace=TRUE)
c <- sample(c(0,5,10),10,replace=TRUE)
df1 <- data.frame(a,b,c)
df1
```

```
##   a  b  c
## 1 1 10 10
## 2 0 0 0
## 3 1 10 0
## 4 5 10 0
## 5 0 10 10
## 6 5 20 10
## 7 5 0 10
## 8 1 20 0
## 9 0 10 10
## 10 1 0 5
```

Na primeira forma, definimos os elementos da matriz, o número de linhas e colunas através dos argumentos `nrow` e `ncol`, e se os elementos serão preenchidos ao longo das linhas (`byrow = TRUE`) ou ao longo das colunas (`byrow = FALSE`).

Na segunda forma, criamos dois vetores e então os juntamos através da função `rbind()`. Repare que na segunda matriz, o R nomeou automaticamente as linhas da matriz com os nomes dos vetores, ao invés de definir seus números.

Para selecionar itens de matrizes também utilizamos os colchetes, primeiro especificando a linha e depois a coluna que desejamos retornar:

```
mat1[2][1]
```

```
## [1] 10
```

4. Data Frames:

- Data frames são estruturas de dados tabulares semelhantes a planilhas ou tabelas de banco de dados.
- Cada coluna de um data frame pode conter um tipo de dado diferente.
- Data frames são amplamente usados para armazenar e manipular conjuntos de dados.
- São frequentemente criados com funções como `data.frame()` ou lidos de arquivos externos.

O **Data Frame** é a estrutura do R utilizada para armazenar elementos em forma de tabela, organizados em linhas e colunas. As colunas e linhas podem ser nomeadas. Você pode criar um data frame com a função `data.frame()`:

```
df1 <- data.frame(c(1,2,3),c("baixo","medio","alto"),c(TRUE, TRUE, FALSE))
print(df1)
```

```
##   c.1..2..3. c..baixo....medio....alto.. c.TRUE..TRUE..FALSE.
## 1           1                   baixo          TRUE
## 2           2                   medio          TRUE
## 3           3                   alto         FALSE
```

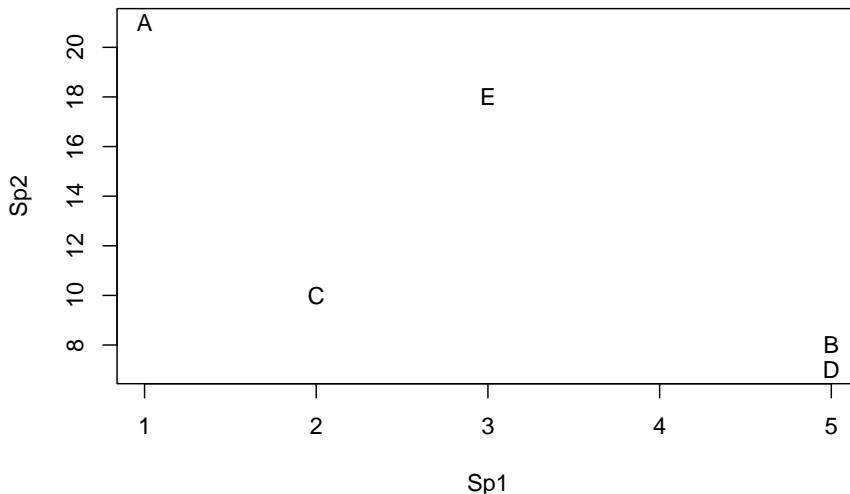
Ou

```
df2 <- data.frame(Sp1=c(1,5,2,5,3), Sp2=c(21,8,10,7,18))
row.names(df2) <- LETTERS[1:5]
df2
```

```
##   Sp1 Sp2
## A   1   21
## B   5    8
## C   2   10
## D   5    7
## E   3   18
```

É possível plotar esse dataframe:

```
plot(df2, type='n'); text(df2, row.names(df2))
```



Dataframes ou Matrizes muito grandes são inviáveis de serem inseridas no ambiente do R através de digitação simples como mostrado anteriormente. Uma opção é usar a função `read.table(text="")` e colar a matriz de dados entre as aspas "..."

```
data <- read.table(text =
Sp1 Sp2 Sp3 Sp4 Sp5 Sp6 Sp7
A 0 0 0 0 0 0 6
```

```
B 0 0 0 2 0 0 10
C 93 2 0 177 0 260 2
D 0 4 0 8 0 0 83
E 0 0 0 0 1 0 0
F 0 0 1 0 0 1 0
G 0 2 0 2 0 0 0
", header = TRUE, row.names = 1)
data
```

```
##   Sp1 Sp2 Sp3 Sp4 Sp5 Sp6 Sp7
## A   0   0   0   0   0   0   6
## B   0   0   0   2   0   0 10
## C  93   2   0 177   0 260   2
## D   0   4   0   8   0   0 83
## E   0   0   0   0   1   0   0
## F   0   0   1   0   0   1   0
## G   0   2   0   2   0   0   0
```

O R vem com alguns conjuntos de dados organizados em Data Frame na sua base, para a finalidade de testes e aprendizado. Um deles é o `mtcars`:

```
print(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
## Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
## Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
## Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
## Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
## Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
## Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
## Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
## Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
## Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
## Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
## Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
## Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1

```

## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87 0 0 3 2
## AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30 0 0 3 2
## Camaro Z28         13.3   8 350.0 245 3.73 3.840 15.41 0 0 3 4
## Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05 0 0 3 2
## Fiat X1-9          27.3   4 79.0 66 4.08 1.935 18.90 1 1 4 1
## Porsche 914-2      26.0   4 120.3 91 4.43 2.140 16.70 0 1 5 2
## Lotus Europa        30.4   4 95.1 113 3.77 1.513 16.90 1 1 5 2
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50 0 1 5 4
## Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50 0 1 5 6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60 0 1 5 8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60 1 1 4 2

```

Algumas funções úteis para conjuntos muito grandes de dados que você deve conhecer:

```
mtcars[1,2]
```

```
## [1] 6
```

```
mtcars["Mazda RX4", "gear"]
```

```
## [1] 4
```

```
mtcars[1:5,1]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7
```

```
mtcars[,2]
```

```
## [1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
```

```
mtcars[3,]
```

```

##           mpg cyl disp hp drat   wt  qsec vs am gear carb
## Datsun 710 22.8   4 108 93 3.85 2.32 18.61  1  1    4    1

```

```
rownames(mtcars)
```

```

## [1] "Mazda RX4"           "Mazda RX4 Wag"      "Datsun 710"
## [4] "Hornet 4 Drive"      "Hornet Sportabout" "Valiant"
## [7] "Duster 360"          "Merc 240D"        "Merc 230"
## [10] "Merc 280"            "Merc 280C"        "Merc 450SE"
## [13] "Merc 450SL"          "Merc 450SLC"      "Cadillac Fleetwood"
## [16] "Lincoln Continental" "Chrysler Imperial" "Fiat 128"
## [19] "Honda Civic"         "Toyota Corolla"   "Toyota Corona"
## [22] "Dodge Challenger"    "AMC Javelin"      "Camaro Z28"

```

```

## [25] "Pontiac Firebird"      "Fiat X1-9"          "Porsche 914-2"
## [28] "Lotus Europa"         "Ford Pantera L"     "Ferrari Dino"
## [31] "Maserati Bora"        "Volvo 142E"

mtcars$gear

## [1] 4 4 4 3 3 3 4 4 4 4 3 3 3 3 3 3 3 3 4 4 4 3 3 3 3 3 4 5 5 5 5 5 4

t(mtcars)

##      Mazda RX4 Mazda RX4 Wag Datsun 710 Hornet 4 Drive Hornet Sportabout
## mpg      21.00      21.000      22.80      21.400      18.70
## cyl       6.00       6.000       4.00       6.000       8.00
## disp     160.00     160.000     108.00     258.000     360.00
## hp      110.00     110.000     93.00     110.000     175.00
## drat      3.90       3.900       3.85       3.080       3.15
## wt       2.62       2.875       2.32       3.215       3.44
## qsec     16.46      17.020      18.61      19.440      17.02
## vs        0.00       0.000       1.00       1.000       0.00
## am       1.00       1.000       1.00       0.000       0.00
## gear      4.00       4.000       4.00       3.000       3.00
## carb      4.00       4.000       1.00       1.000       2.00
##      Valiant Duster 360 Merc 240D Merc 230 Merc 280 Merc 280C Merc 450SE
## mpg      18.10      14.30      24.40      22.80      19.20      17.80      16.40
## cyl       6.00       8.00       4.00       4.00       6.00       6.00       8.00
## disp    225.00     360.00     146.70     140.80     167.60     167.60     275.80
## hp      105.00     245.00     62.00      95.00     123.00     123.00     180.00
## drat      2.76       3.21       3.69       3.92       3.92       3.92       3.07
## wt       3.46       3.57       3.19       3.15       3.44       3.44       4.07
## qsec     20.22      15.84      20.00      22.90      18.30      18.90      17.40
## vs        1.00       0.00       1.00       1.00       1.00       1.00       0.00
## am       0.00       0.00       0.00       0.00       0.00       0.00       0.00
## gear      3.00       3.00       4.00       4.00       4.00       4.00       3.00
## carb      1.00       4.00       2.00       2.00       4.00       4.00       3.00
##      Merc 450SL Merc 450SLC Cadillac Fleetwood Lincoln Continental
## mpg      17.30      15.20          10.40      10.400
## cyl       8.00       8.00           8.00       8.000
## disp    275.80     275.80          472.00     460.000
## hp      180.00     180.00          205.00     215.000
## drat      3.07       3.07          2.93       3.000
## wt       3.73       3.78          5.25       5.424
## qsec     17.60      18.00          17.98      17.820
## vs        0.00       0.00           0.00       0.000
## am       0.00       0.00           0.00       0.000
## gear      3.00       3.00           3.00       3.000

```

```

## carb      3.00      3.00      4.00      4.000
## Chrysler Imperial Fiat 128 Honda Civic Toyota Corolla Toyota Corona
## mpg        14.700    32.40    30.400    33.900    21.500
## cyl         8.000     4.00     4.000     4.000     4.000
## disp      440.000    78.70    75.700    71.100   120.100
## hp       230.000    66.00    52.000    65.000    97.000
## drat       3.230     4.08     4.930     4.220     3.700
## wt          5.345     2.20     1.615     1.835     2.465
## qsec      17.420    19.47    18.520    19.900   20.010
## vs           0.000     1.00     1.000     1.000     1.000
## am           0.000     1.00     1.000     1.000     0.000
## gear        3.000     4.00     4.000     4.000     3.000
## carb        4.000     1.00     2.000     1.000     1.000
## Dodge Challenger AMC Javelin Camaro Z28 Pontiac Firebird Fiat X1-9
## mpg        15.50     15.200   13.30     19.200   27.300
## cyl          8.00     8.000    8.00     8.000     4.000
## disp      318.00    304.000  350.00   400.000   79.000
## hp       150.00    150.000  245.00   175.000   66.000
## drat       2.76      3.150   3.73     3.080   4.080
## wt          3.52      3.435   3.84     3.845   1.935
## qsec      16.87     17.300   15.41    17.050   18.900
## vs           0.00     0.000    0.00     0.000   1.000
## am           0.00     0.000    0.00     0.000   1.000
## gear        3.00      3.000    3.00     3.000   4.000
## carb        2.00      2.000    4.00     2.000   1.000
## Porsche 914-2 Lotus Europa Ford Pantera L Ferrari Dino Maserati Bora
## mpg        26.00     30.400   15.80     19.70   15.00
## cyl          4.00     4.000    8.00     6.00    8.00
## disp      120.30    95.100  351.00   145.00  301.00
## hp       91.00     113.000  264.00   175.00  335.00
## drat       4.43      3.770   4.22     3.62    3.54
## wt          2.14      1.513   3.17     2.77    3.57
## qsec      16.70     16.900   14.50     15.50  14.60
## vs           0.00     1.000    0.00     0.00   0.00
## am           1.00     1.000    1.00     1.00   1.00
## gear        5.00      5.000    5.00     5.00   5.00
## carb        2.00      2.000    4.00     6.00   8.00
## Volvo 142E
## mpg        21.40
## cyl          4.00
## disp      121.00
## hp       109.00
## drat       4.11
## wt          2.78
## qsec      18.60
## vs           1.00

```



```

## [136] 3.690 3.920 3.920 3.920 3.070 3.070 3.070 2.930 3.000
## [145] 3.230 4.080 4.930 4.220 3.700 2.760 3.150 3.730 3.080
## [154] 4.080 4.430 3.770 4.220 3.620 3.540 4.110 2.620 2.875
## [163] 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440
## [172] 4.070 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835
## [181] 2.465 3.520 3.435 3.840 3.845 1.935 2.140 1.513 3.170
## [190] 2.770 3.570 2.780 16.460 17.020 18.610 19.440 17.020 20.220
## [199] 15.840 20.000 22.900 18.300 18.900 17.400 17.600 18.000 17.980
## [208] 17.820 17.420 19.470 18.520 19.900 20.010 16.870 17.300 15.410
## [217] 17.050 18.900 16.700 16.900 14.500 15.500 14.600 18.600 1.000
## [226] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
## [235] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
## [244] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 4.000
## [253] 4.000 4.000 3.000 3.000 3.000 3.000 4.000 4.000 4.000
## [262] 4.000 3.000 3.000 3.000 3.000 3.000 3.000 4.000 4.000
## [271] 4.000 3.000 3.000 3.000 3.000 3.000 4.000 5.000 5.000
## [280] 5.000 5.000 5.000 4.000 4.000 4.000 1.000 1.000 2.000
## [289] 1.000 4.000 2.000 2.000 4.000 4.000 3.000 3.000 3.000
## [298] 4.000 4.000 4.000 1.000 2.000 1.000 1.000 2.000 2.000
## [307] 4.000 2.000 1.000 2.000 2.000 4.000 6.000 8.000 2.000

length(as.matrix(mtcars))

## [1] 352

dim(mtcars)

## [1] 32 11

length(mtcars[mtcars==0])

## [1] 37

str(mtcars)

## 'data.frame': 32 obs. of 11 variables:
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num 160 160 108 258 360 ...
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num 16.5 17 18.6 19.4 17 ...
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...

```

```

## $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num  4 4 1 1 2 1 4 2 2 4 ...

summary(mtcars)

##      mpg          cyl          disp          hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   :52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.:96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##      drat         wt          qsec          vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50  Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89  1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71  Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85  Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90  3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90  Max.   :1.0000
##      am          gear         carb
##  Min.   :0.0000  Min.   :3.000  Min.   :1.000
##  1st Qu.:0.0000  1st Qu.:3.000  1st Qu.:2.000
##  Median :0.0000  Median :4.000  Median :2.000
##  Mean   :0.4062  Mean   :3.688  Mean   :2.812
##  3rd Qu.:1.0000  3rd Qu.:4.000  3rd Qu.:4.000
##  Max.   :1.0000  Max.   :5.000  Max.   :8.000

```

As funções `str()` e `summary()` trazem algumas informações úteis sobre o data frame, de uma forma geral. É comum utilizá-las ao importar uma base de dados para o R, para se ter uma ideia do que está sendo analisado. O `str()` conta as linhas e colunas e mostra os seus tipos e alguns valores de exemplo. Já o `summary()` traz médias, medianas, máximos, mínimos e quantidade de valores indisponíveis em cada coluna.

5. Fatores:

- Fatores são usados para representar variáveis categóricas ou de fatores.
- São úteis para análises estatísticas e gráficos.
- Podem ser criados com a função `factor()`.

Por fim, temos os fatores (**factors**). Fatores são usados para representar categorias. Isso quer dizer que em uma variável do tipo `factor`, aquele valor terá uma quantidade limitada de valores. Pense, por exemplo, nos meses do ano ou letras do alfabeto. Existe um número definido de meses, e o valor estará dentro destas possibilidades. Um nome de uma pessoa, ao contrário, tem infinitas possibilidades, pois não existe uma lista pré-definida de nomes de pessoas, e qualquer um pode criar um novo nome. Para criar um fator, vamos aproveitar

o exemplo dos meses do ano e utilizar a função `factor()`:

```
meses <- c("Jan", "Jan", "Feb", "Mar", "Apr", "Apr", "May", "Jan", "Jan", "May", "May", "Jun")
meses
## [1] "Jan" "Jan" "Feb" "Mar" "Apr" "Apr" "May" "Jan" "Jan" "May" "May" "Jun"

meses <- factor(meses)
print(meses)

## [1] Jan Jan Feb Mar Apr Apr May Jan Jan May May Jun
## Levels: Apr Feb Jan Jun Mar May

table(meses)

## meses
## Apr Feb Jan Jun Mar May
## 2   1   4   1   1   3
```

Um `vector` ou “vetor” é uma estrutura de dados fundamental que armazena uma sequência ordenada de elementos do mesmo tipo. Ao imprimir (`print()`) o vetor `meses`, vemos que o R nos indica os níveis (`levels`), que nada mais são do que os valores únicos que existem naquele vetor. Usando a função `table()`, podemos ver a contagem de cada uma das categorias. Vemos que elas estão ordenadas em ordem alfabética.

Como isso pode ser aplicado?

```
meses <- c("Jan", "Jan", "Jan", "Feb", "Feb", "Feb", "Feb", "Mar", "Mar", "Mar", "Apr", "Apr")
meses
## [1] "Jan" "Jan" "Jan" "Feb" "Feb" "Feb" "Feb" "Mar" "Mar" "Mar" "Apr" "Apr"
## [13] "Apr" "May" "May" "May" "Jun" "Jun" "Jun"

factor(meses)

## [1] Jan Jan Jan Feb Feb Feb Mar Mar Mar Apr Apr Apr May May May Jun Jun Jun
## Levels: Apr Feb Jan Jun Mar May

print(meses)

## [1] "Jan" "Jan" "Jan" "Feb" "Feb" "Feb" "Feb" "Mar" "Mar" "Mar" "Apr" "Apr"
## [13] "Apr" "May" "May" "May" "Jun" "Jun" "Jun"

table(meses)

## meses
```

```
## Apr Feb Jan Jun Mar May
## 3 4 3 3 3 3

riq <- c(3, 5, 7, 6, 8, 9, 0, 4, 6, 9, 2, 6, 3, 9, 6, 0, 1, 6, 7)
dados <- data.frame(Mes = meses, Riqueza = riq)
riq

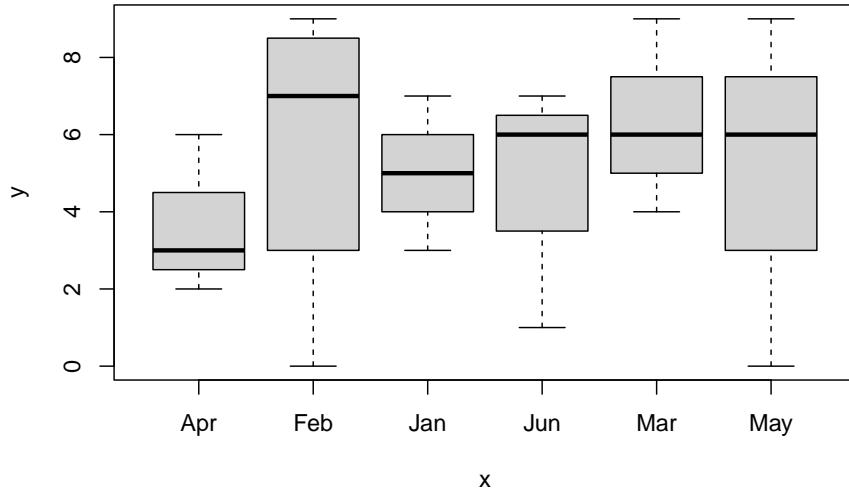
## [1] 3 5 7 6 8 9 0 4 6 9 2 6 3 9 6 0 1 6 7
dados

##      Mes Riqueza
## 1 Jan      3
## 2 Jan      5
## 3 Jan      7
## 4 Feb      6
## 5 Feb      8
## 6 Feb      9
## 7 Feb      0
## 8 Mar      4
## 9 Mar      6
## 10 Mar     9
## 11 Apr     2
## 12 Apr     6
## 13 Apr     3
## 14 May     9
## 15 May     6
## 16 May     0
## 17 Jun     1
## 18 Jun     6
## 19 Jun     7

factor(meses)

## [1] Jan Jan Jan Feb Feb Feb Mar Mar Mar Apr Apr Apr May May May Jun Jun Jun
## Levels: Apr Feb Jan Jun Mar May

plot(factor(meses), riq)
```



```
by(riq, meses, mean)
```

```
## meses: Apr
## [1] 3.666667
## -----
## meses: Feb
## [1] 5.75
## -----
## meses: Jan
## [1] 5
## -----
## meses: Jun
## [1] 4.666667
## -----
## meses: Mar
## [1] 6.333333
## -----
## meses: May
## [1] 5
```

Uma situação diferente ocorre quando tem-se colunas com o mesmo nome.

```
df <- data.frame(
  A = c(1, 2, 3),
```

```

B = c(4, 5, 6),
A = c(7, 8, 9), #nome duplicado
C = c(0, 0, 1),
D = c(1, 1, 0),
E = c(1, 0, 0),
F = c(11, 12, 13),
G = c(1, 2, 0),
E = c(2, 1, 1), #nome duplicado
check.names = FALSE
)
df

```

```

##   A B A C D E   F G E
## 1 1 4 7 0 1 1 11 1 2
## 2 2 5 8 0 1 0 12 2 1
## 3 3 6 9 1 0 0 13 0 1

```

Por exemplo, nesse dataframe temos colunas duplicadas. Nesse caso, encontrar e resolver colunas com o mesmo nome é crucial para garantir a integridade dos dados e realizar uma análise precisa. Você deve consolidar essas colunas duplicadas, somando-as ou fazendo sua média.

```

# Achando colunas com nomes duplicados
dup_cols <- names(df)[duplicated(names(df))]
# Somando colunas com o mesmo nome
for (col_name in unique(dup_cols)) {
  # Get indices of columns with the same name
  col_indices <- which(names(df) == col_name)
  # Sum columns with the same name
  df[[col_name]] <- rowSums(df[, col_indices, drop = FALSE])
}
# Remove as colunas duplicadas originais e mantem as novas colunas que são a soma ("except for the
df <- df[, !duplicated(names(df))]
# Mostra a nova tabela com colunas repetidas somadas
print(df)

```

```

##   A B C D E   F G
## 1 8 4 0 1 3 11 1
## 2 10 5 0 1 1 12 2
## 3 12 6 1 0 1 13 0

```

6. Listas

Listas são um tipo especial de vetor, que podem conter elementos de diferentes tipos, incluindo vetores. Veja a lista baseada em alguns vetores, a seguir:

```
a <- c(3,6,9)
b <- c("a","b","c","d")
c <- c(TRUE, FALSE, TRUE, TRUE)
lista1 <- list(a,b,c)
print(lista1)
```

```
## [[1]]
## [1] 3 6 9
##
## [[2]]
## [1] "a" "b" "c" "d"
##
## [[3]]
## [1] TRUE FALSE TRUE TRUE
```

RESUMO (De <http://venus.ifca.unican.es/Rintro/dataStruct.html>)

- Vetores: arrays unidimensionais usados para armazenar coleções de dados do mesmo tipo

Vetores Numéricos (mode: numeric)

Vetores Complexos (mode: complex)

Vetores Lógicos (mode: logical)

Vetor de Caracteres ou strings de texto (mode: character)

- Matrizes: arrays bidimensionais para armazenar coleções de dados do mesmo modo. São acessados por dois índices inteiros.
- Arrays: semelhantes a matrizes, mas podem ser multidimensionais (mais de duas dimensões).
- Fatores: vetores de variáveis categóricas projetados para agrupar os componentes de outro vetor com o mesmo tamanho.
- Listas: coleção ordenada de objetos, onde os elementos podem ser de tipos diferentes.
- Data Frames: generalização de matrizes onde diferentes colunas podem armazenar dados de modo diferente.
- Funções: objetos criados pelo usuário e reutilizados para realizar operações específicas.

7.7 Como pedir ajuda?

7.7.1 Dentro do R

No R, você pode solicitar ajuda de várias maneiras:

1. `help()` ou `?`:

Data Type	Example	Description					
Logical	TRUE, FALSE	boolean values					
Numeric	2, 45.9, 3782	Numbers of all kinds					
Integer	9L, 779L	Explicitly Integers					
Complex	8+9i	Real Value + Complex Value					
Character	'm', "hello"	Characters and Strings					
Raw	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>68</td><td>65</td><td>6C</td><td>6C</td><td>6F</td></tr> </table> is the value for hello	68	65	6C	6C	6F	Any data is stored as raw bytes
68	65	6C	6C	6F			

Figure 7.2: Resumo visual dos tipos de dados no ambiente de programação R.
 (De: <https://www.tutorialkart.com/r-tutorial/r-data-types>)

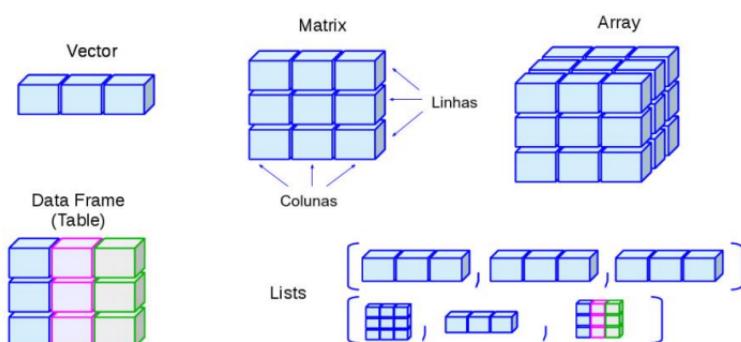


Figure 7.3: Resumo visual dos tipos de dados no ambiente de programação R.
 (De: <https://lhmet.github.io/adar-ebook/estrutura-dados.html>)

Você pode usar a função `help()` seguida pelo nome da função ou pacote sobre o qual você precisa de ajuda. Por exemplo:

```
help(mean)
```

Ou, de forma mais concisa:

```
?mean
```

2. `help.search()`:

Você pode pesquisar por termos relacionados à sua dúvida usando a função `help.search()`.

```
help.search("linear regression")
```

3. `example()`:

Para ver exemplos de como usar uma função, você pode usar a função `example()`.

```
example(mean)
```

7.7.2 Fora do R

4. R Site e Documentação Online:

O site oficial do R (<https://www.r-project.org/>) oferece uma vasta documentação, tutoriais e recursos para ajudá-lo a aprender e resolver problemas.

5. Fóruns e Comunidades Online:

Existem várias comunidades online onde você pode fazer perguntas e obter ajuda de outros usuários do R, como o Stack Overflow (<https://stackoverflow.com/>), o RStudio Community (<https://community.rstudio.com/>), entre outros.

6. Livros e Tutoriais:

Existem muitos livros e tutoriais disponíveis que podem ajudá-lo a aprender R, desde os conceitos básicos até técnicas avançadas. Os livros em Português mais importantes são FERREIRA; OLIVEIRA (2020).

Ao usar esses recursos, certifique-se de fornecer informações claras sobre sua dúvida ou problema, incluindo detalhes como o código que você está usando, mensagens de erro (se houver) e o que você já tentou fazer para resolver o problema. Isso ajudará os outros a entenderem melhor sua situação e fornecer uma resposta mais útil.

7.7.3 ChatGPT

Pergunte ao ChatGPT <https://chat.openai.com/>

Para usar o ChatGPT para pedir ajuda ao R, você pode solicitar assistência na elaboração de código, depurar problemas, ou mesmo pedir orientação sobre como realizar uma determinada tarefa em R. Aqui está um exemplo de como você pode pedir ajuda ao R:

Oi ChatGPT! Estou enfrentando um problema ao tentar realizar uma análise de regressão linear no R. Eu tenho um conjunto de dados chamado "dados.csv" e estou tentando ajustar um modelo linear simples com a função lm(). No entanto, estou recebendo um erro que não consigo entender. Você poderia me ajudar a resolver isso?

Ao fazer uma pergunta específica e fornecer detalhes sobre o que você está tentando alcançar e quais problemas está enfrentando, o ChatGPT pode oferecer orientações úteis e sugestões para ajudá-lo com o seu código em R.

Detalhe, o texto acima foi gerado pelo próprio ChatGPT (Figura 7.4). Você pode chamar o ChatGPT pelo próprio R:

```
# Abrir o link para o site do ChatGPT
browseURL("https://chat.openai.com/")
```

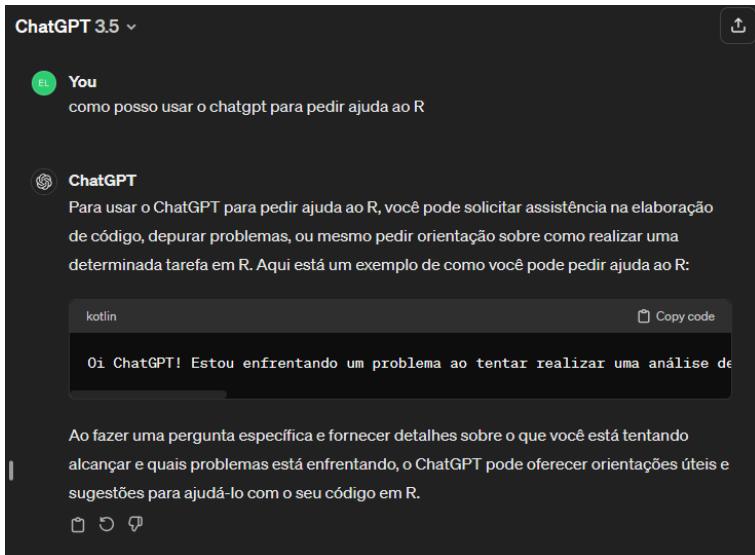


Figure 7.4: Usando o ChatGPT para pedir ajuda sobre o R.

7.8 Work flow

Você já deve estar percebendo que a programação em R pode se tornar confusa, pela quantidade de possibilidades disponíveis e formas diferentes de se obter o mesmo resultado.

Por isso é necessário que o usuário tenha um workflow que permita o reconhecimento de uma **sequência lógica para estruturar e aplicar** os devidos testes estatísticos e construção de gráficos e tabelas, após a organização e manipulação dos dados. Uma sugestão de workflow preliminar e vista na figura 7.5 (modificada de OLIVEIRA; GUERRA; McDONNELL (2018)):

- Carregar os dados
- Limpar os dados
- Transformar, visualizar e modelar (fase exploratória)
- Comunicar o resultado

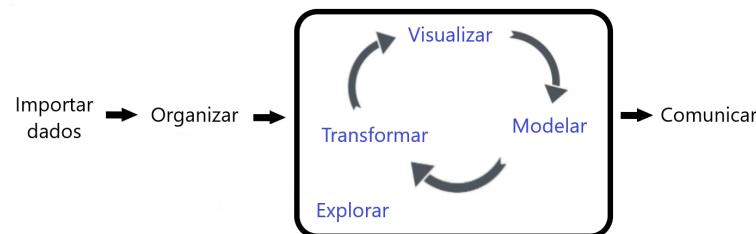


Figure 7.5: Exemplo de workflow para análises no R (modificada de [@RN2941])

7.9 Reconhecendo a estrutura dos dados do R em uma matriz multivariada

7.9.1 Organização básica

```
rm(list=ls(all=TRUE)) #limpa a memória
```

Instalando os pacotes necessários para esse módulo

```
install.packages("openxlsx") #importa arquivos do excel
```

```
library(openxlsx)
```

Os códigos acima, são usados para instalar e carregar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades

do R (R CRAN (TEAM, 2017) e RStudio) (TEAM, 2022). No exemplo acima, o pacote `openxlsx` permite ler e escrever arquivos Excel no R. Para instalar um pacote no R, você precisa usar a função `install.packages()`.

Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`. Por exemplo, para carregar o pacote `openxlsx`, você precisa executar a função `library(openxlsx)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

Usaremos uma matriz multivariada (sítios x espécies, matriz comunitária) do Projeto PPBio chamada `ppbio**.xlsx` que está no diretório “C:/Meu/Diretório/De/Trabalho/Planilha.xlsx”

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações.

Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

Definindo o diretório de trabalho e installando os pacotes necessários:

```
getwd()  
setwd("C:/Seu/Diretório/De/Trabalho")
```

O símbolo `?` é usado para acessar a documentação de uma função ou um pacote no R. Como mostrado acima você pode saber mais sobre a função `getwd()`, usando o comando `?getwd`. Isso vai abrir uma página no menu de ajuda com a descrição, os argumentos, os valores de retorno e os exemplos da função `getwd()`. Você também pode usar o símbolo `?` para obter informações sobre um pacote inteiro. Por exemplo, se você quiser saber mais sobre o pacote `openxlsx`, você pode digitar `?openxlsx`. Isso vai abrir uma página com a visão geral, a instalação, os recursos e as referências do pacote solicitado.

7.10 Importando a planilha

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações. Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

```

library(openxlsx)
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06c-peis.xlsx",
                     rowNames = T,
                     colNames = T,
                     sheet = "Sheet1")
ppbio[1:10, 1:10] #primeiras 10 linhas e 10 colunas da matriz

##          ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
## S-A-ZA1      0       1       0       0       0       0       0       0
## S-R-CC1      0      99       0       0       0       0       0       0
## S-R-CT1      0     194      55       0       0       5       0       1
## S-R-CP1      0      19       0       0       0       0       0       0
## S-A-TA1      0      23       1      13       0       0       0       0
## S-R-CT2      0     142       3       3       0      69       0       0
## S-R-CP2      0       5       1       0      40       9       0       0
## S-A-TA2      0      46       0     178       0       0       0       0
## S-R-CT3      0     206      64       0       0      25       0       0
## S-R-CP3      0      16       0       0      13      24       0       0
##          cr-menez cu-lepid
## S-A-ZA1      0       0
## S-R-CC1      0       0
## S-R-CT1     14       0
## S-R-CP1      0       0
## S-A-TA1      0       0
## S-R-CT2      4       0
## S-R-CP2      0       0
## S-A-TA2      0       0
## S-R-CT3      8       0
## S-R-CP3      0       0

```

Atente para os resultados dos comandos a seguir.

```

#View(ppbio)
print(ppbio[1:10, 1:10])
#ppbio
str(ppbio)
#?str
mode(ppbio)
#?mode
class(ppbio)
#?class
ppbio_ma <- as.matrix(ppbio) #lê ppbio como uma matriz
str(ppbio_ma)

```

```
##          ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
```

```

## S-A-ZA1      0      1      0      0      0      0      0      0      0
## S-R-CC1      0     99      0      0      0      0      0      0      0
## S-R-CT1      0    194     55      0      0      0      5      0      1
## S-R-CP1      0     19      0      0      0      0      0      0      0
## S-A-TA1      0     23      1     13      0      0      0      0      0
## S-R-CT2      0    142      3      3      0      0     69      0      0
## S-R-CP2      0      5      1      0     40      9      0      0      0
## S-A-TA2      0     46      0    178      0      0      0      0      0
## S-R-CT3      0    206     64      0      0      0     25      0      0
## S-R-CP3      0     16      0      0     13     24      0      0      0
##           cr-menez cu-lepid
## S-A-ZA1      0      0
## S-R-CC1      0      0
## S-R-CT1     14      0
## S-R-CP1      0      0
## S-A-TA1      0      0
## S-R-CT2      4      0
## S-R-CP2      0      0
## S-A-TA2      0      0
## S-R-CT3      8      0
## S-R-CP3      0      0
## 'data.frame':   26 obs. of  35 variables:
## $ ap-davis : num  0 0 0 0 0 0 0 0 0 ...
## $ as-bimac : num  1 99 194 19 23 142 5 46 206 16 ...
## $ as-fasci : num  0 0 55 0 1 3 1 0 64 0 ...
## $ ch-bimac : num  0 0 0 0 13 3 0 178 0 0 ...
## $ ci-ocela : num  0 0 0 0 0 40 0 0 13 ...
## $ ci-orien : num  0 0 5 0 0 69 9 0 25 24 ...
## $ co-macro : num  0 0 0 0 0 0 0 0 0 ...
## $ co-heter : num  0 0 1 0 0 0 0 0 0 ...
## $ cr-menez : num  0 0 14 0 0 4 0 0 8 0 ...
## $ cu-lepid : num  0 0 0 0 0 0 0 0 0 ...
## $ cy-gilbe : num  0 0 0 0 0 0 0 0 0 ...
## $ ge-brasi : num  0 0 3 0 0 0 0 0 1 0 ...
## $ he-margi : num  0 0 0 0 0 1 0 0 0 0 ...
## $ ho-malab : num  0 0 1 5 0 17 10 2 31 4 ...
## $ hy-pusar : num  0 0 9 2 0 43 2 0 11 0 ...
## $ le-melan : num  0 0 0 0 0 0 0 0 0 ...
## $ le-piau : num  0 0 3 0 0 1 3 0 2 1 ...
## $ le-taeni : num  0 0 0 0 0 0 0 0 0 ...
## $ mo-costa : num  0 0 0 0 0 0 0 0 0 ...
## $ mo-lepid : num  0 1 39 0 0 1 0 0 0 0 ...
## $ or-nilot : num  0 2 36 0 0 77 0 0 138 0 ...
## $ pa-manag : num  0 0 0 0 0 0 0 0 0 ...
## $ pimel-sp : num  0 0 6 0 0 0 0 0 0 ...
## $ po-retic : num  0 0 0 0 20 0 0 5 0 ...

```

```

## $ po-vivip : num  0 0 47 15 0 221 32 0 326 10 ...
## $ pr-brevi : num  9 0 5 0 1 15 5 2 164 0 ...
## $ ps-rhomb : num  0 0 0 0 0 0 0 0 1 0 ...
## $ ps-genise: num  0 0 0 0 0 0 0 0 1 0 ...
## $ se-heter : num  0 0 40 14 4 60 0 0 38 0 ...
## $ se-piaba : num  0 0 68 0 0 0 0 0 0 0 ...
## $ se-spilo : num  0 0 0 0 0 0 0 0 1 0 ...
## $ st-noton : num  0 0 1 0 0 25 0 0 115 0 ...
## $ sy-marmo : num  0 0 0 0 0 0 1 0 0 0 ...
## $ te-chalc : num  0 0 0 0 0 0 0 0 0 0 ...
## $ tr-signa : num  0 0 18 0 0 15 0 0 7 0 ...
## [1] "list"
## [1] "data.frame"
## num [1:26, 1:35] 0 0 0 0 0 0 0 0 0 0 ...
## - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
##   ..$ : chr [1:35] "ap-davis" "as-bimac" "as-fasci" "ch-bimac" ...

```

Apêndices

Snippet e chunk

Em programação R, tanto **snippet** (“trecho de código”) quanto **chunk** “pedaço de código” referem-se a porções de código usadas dentro de um contexto maior, geralmente no contexto de sessões interativas ou documentos estruturados como R Markdown. Aqui está como eles diferem:

1. Snippet:

- Um “trecho de código” em R geralmente se refere a um pedaço pequeno e isolado de código que executa uma tarefa ou operação específica.
- Trechos de código são frequentemente usados de forma interativa, seja em um console R ou dentro de um ambiente de desenvolvimento integrado (IDE) como o RStudio.
- Esses trechos são geralmente segmentos curtos de código usados para experimentação, testes ou cálculos rápidos.

2. Chunk:

- Um “trecho” em R é um termo comumente usado no contexto da programação literária, especialmente em documentos escritos usando R Markdown.
- Trechos são blocos maiores de código que podem ser executados independentemente dentro de um documento R Markdown.
- Os trechos são delimitados por delimitadores especiais, como `{r}`, que indicam ao processador R Markdown que o conteúdo incluído é código R.
- Documentos R Markdown permitem a integração de trechos de

código R com texto narrativo, permitindo que os usuários combinem código, resultados e texto explicativo em um único documento.

Em resumo, enquanto um “trecho de código” geralmente se refere a uma pequena e isolada peça de código usada para testes ou experimentação, um “trecho” se refere a um bloco maior de código usado dentro do contexto da programação literária, particularmente em documentos escritos usando R Markdown. Os trechos permitem a integração de código e texto narrativo em documentos estruturados.

Sites consultados

Recomendo visitar esses sites e ver demais conteúdos relacionados

Visite:

<https://felipegalvao.com.br/pt/blog/basic-r-introduction-data-types-and-structures/>
<https://lhmet.github.io/adar-ebook/datatype.html>

Vídeos:

Objetos, atributos e tipos de dados em R: <https://www.youtube.com/watch?v=YdUUQapPYzs&t=1s>

Outros materiais de apoio:

Tipos de dados e operadores I (Parte 8): <https://www.youtube.com/watch?v=Dy2cI02WbMQ>

Tipos de dados e operadores II (Parte 9) <https://www.youtube.com/watch?v=E6ZTBbicTvg>

Estrutura de dados I (Parte 10): <https://www.youtube.com/watch?v=xzlM33Tqvic>

Estrutura de dados I (Parte 11): <https://www.youtube.com/watch?v=aEaSzIDMcuI>

Script limpo

Aqui apresento o script na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executá-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

```
"Início do R Módulo"
600+66
a <- 5 + 3
a
print(a)
class(a)
var1 <- 3
```

```
var2 <- as.integer(3)
class(var1)
class(var2)
b <- "Ola mundo"
c <- TRUE
class(b)
class(c)
militares <- c("Soldado", "Soldado", "Coronel", "General", "Tenente", "Tenente", "Cabo", "Soldado")
militares
militares <- factor(militares)
print(militares)
table(militares)
var1 <- 3
is.vector(var1)
var1 <- c(3, 6, 7.8, 332)
print(var1)
var2 <- c("Ola", "tudo", "bem")
print(var2)
var3 <- c(TRUE, FALSE, TRUE, TRUE, FALSE, FALSE, TRUE, TRUE, TRUE)
print(var3)
mat1 <- matrix(
  c(1, 5, 10, 30, 15, 8),
  nrow=3,
  ncol=2,
  byrow=TRUE)
print(mat1)
vec1 <- c(1, 5)
vec2 <- c(10, 30)
vec3 <- c(15, 8)
mat2 <- rbind(vec1, vec2, vec3)
print(mat2)
class(mat2)
mat1[2] [1]
a <- c(3, 6, 9)
b <- c("a", "b", "c", "d")
c <- c(TRUE, FALSE, TRUE, TRUE)
list1 <- list(a, b, c)
print(list1)
df1 <- data.frame(c(1, 2, 3), c("baixo", "medio", "alto"), c(TRUE, TRUE, FALSE))
print(df1)
print(mtcars)
mtcars[1, 2]
mtcars["Mazda RX4", "gear"]
mtcars[1:5, 1]
```

```
mtcars[,2]
mtcars[3,]
nrow(mtcars)
ncol(mtcars)
dim(mtcars)
str(mtcars)
summary(mtcars)
## rm(list=ls(all=TRUE)) #limpa a memória
## install.packages("openxlsx") #importa arquivos do excel
library(openxlsx)
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
library(openxlsx)
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/bentos06.xlsx",
                    rowNames = T,
                    colNames = T,
                    sheet = "contagem")
ppbio[1:10, 1:10] #primeiras 10 linhas e 10 colunas da matriz
#View(ppbio)
print(ppbio[1:10, 1:10])
#ppbio
str(ppbio)
#?str
mode(ppbio)
#?mode
class(ppbio)
#?class
ppbio_ma <- as.matrix(ppbio) #lê ppbio como uma matriz
str(ppbio_ma)
```

Referências

Chapter 8

R Modulo 3.1 - Estatísticas descritivas e normalidade

RESUMO

A estatística descritiva tem um papel importante a desempenhar na ciência. Quando problemas específicos são tratados na ciência, os dados precisam ser coletados, analisados e apresentados de forma concisa para que outros possam se beneficiar do que foi encontrado.

Apresentação

A estatística descritiva tem um papel importante a desempenhar na ciência. Quando problemas específicos são tratados na ciência, os dados precisam ser coletados, analisados e apresentados de forma concisa para que outros possam se beneficiar do que foi encontrado. Geralmente não é possível apresentar um conjunto de dados completo em uma publicação ou em um seminário e, mesmo que fosse, é improvável isso permitisse uma boa comunicação dos resultados da pesquisa. Em vez disso, os dados são geralmente resumidos como tabelas de frequência, histogramas e estatísticas descritivas que os leitores ou ouvintes podem assimilar prontamente, mas que ainda transmitem os elementos essenciais do conjunto de dados original. O principal objetivo do cálculo das estatísticas descritivas é transmitir informações essenciais contidas em um conjunto de dados da forma mais concisa e clara possível.

8.1 Sobre os dados

Considere os dados sobre o comprimento dos brotos de *Banksia ericifolia* de charnecas¹ na Baía de Jervis, Austrália (BRADSTOCK; TOZER; KEITH, 1997) (Figura 8.1). Existem 500 medições, um conjunto de dados formidável. Por inspeção, o comprimento mínimo do broto é 10,0 e o máximo é 44,9 cm. Esses valores definem o intervalo da amostra. Agora precisamos subdividir o intervalo em intervalos ou classes, cada um com o mesmo tamanho. Geralmente, é aconselhável arredondar o valor mínimo para baixo e o valor máximo para valores apropriados ao decidir as classes de intervalos. Nesse caso, parece sensato dividir a faixa de 10 a 45 cm em sete intervalos a cada 5 cm de largura. Se contarmos o número de brotos que se encontram em cada um dos sete intervalos, temos a base para a tabulação da frequência. A coluna de frequência foi obtida contando o número de medições que existem dentro de cada classe. A coluna de frequência percentual foi obtida representando cada contagem como uma porcentagem da contagem total. A frequência cumulativa e as frequências percentuais cumulativas foram obtidas somando progressivamente as frequências correspondentes.

8.2 Organização básica

```
rm(list=ls(all=TRUE)) #limpa a memória
```

Instalando os pacotes necessários para esse módulo

```
install.packages("openxlsx") #importa arquivos do excel
install.packages("fdth")
```

```
library(openxlsx)
```

Os códigos acima, são usados para instalar e carregar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades do R (R CRAN (TEAM, 2017) e RStudio (TEAM, 2022)). No exemplo acima, o pacote `openxlsx` permite ler e escrever arquivos Excel no R. Para instalar um pacote no R, você precisa usar a função `install.packages()`.

Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`. Por exemplo, para carregar o pacote `openxlsx`, você precisa executar a função `library(openxlsx)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado.

¹A etimologia do gênero *Astyanax* vem da mitologia Grega. Heitor personagem da “Ilíada”, tinha um filho chamado Astianax.

28.4	29.1	25.7	26.2	25.8	30.2	26.2	35.8	33.4	29.0	31.6	33.8	22.6	34.8
25.1	36.8	29.9	38.3	27.1	32.3	29.0	27.7	28.1	28.9	21.8	25.0	23.2	26.8
29.9	36.3	26.0	21.2	19.8	36.7	21.1	34.6	29.6	32.6	27.2	34.2	26.7	27.6
25.8	23.2	28.8	38.2	32.7	38.7	33.2	24.5	21.6	28.6	19.4	27.4	43.7	25.5
35.0	25.1	24.7	29.5	24.9	29.2	19.5	20.1	30.3	38.9	28.2	26.2	29.4	22.4
36.4	31.8	31.0	39.4	28.8	31.8	28.7	37.0	25.5	19.3	44.0	38.0	28.6	36.5
29.1	21.1	30.4	31.2	38.0	39.0	19.3	27.6	19.1	32.5	26.8	39.9	36.1	41.5
33.2	26.5	38.1	14.9	33.2	27.8	24.7	24.9	25.0	33.1	24.1	19.7	19.1	31.4
26.9	22.5	25.5	33.0	19.4	26.8	24.6	37.5	19.8	43.7	38.1	30.8	34.5	22.8
34.2	33.6	42.5	19.0	25.8	34.0	34.4	42.0	35.4	31.5	40.6	19.2	24.9	33.5
32.0	38.4	29.1	29.4	29.3	26.8	32.4	25.2	28.5	29.8	22.8	17.1	29.6	33.3
31.7	22.4	21.7	20.1	21.6	23.5	33.2	33.0	29.6	36.9	26.8	38.1	29.8	21.2
23.6	16.2	27.3	33.3	21.6	30.2	22.5	33.0	38.3	29.5	34.9	30.3	26.0	24.5
31.1	31.7	31.6	41.7	25.9	32.3	35.7	31.6	26.0	26.6	26.3	19.6	22.0	40.2
29.1	15.8	22.1	23.2	25.4	28.4	20.2	25.5	26.9	32.7	28.8	39.6	31.9	31.9
29.8	27.1	36.9	32.7	24.8	18.0	40.2	28.0	26.8	41.9	15.8	16.2	33.6	34.8
31.5	27.4	37.2	30.6	32.2	34.8	28.2	31.3	34.3	32.0	33.3	30.1	20.5	37.3
20.6	27.2	25.0	26.1	34.5	44.9	40.5	32.1	40.4	35.7	33.9	29.3	28.1	34.3
29.3	24.7	37.0	36.9	34.8	29.8	22.8	32.3	34.8	29.6	33.6	22.8	31.6	34.7
24.0	30.5	31.6	28.7	20.8	14.6	23.4	26.3	31.9	32.5	34.3	25.7	36.0	37.7
32.4	36.4	24.1	33.1	26.3	35.7	26.4	34.7	27.5	39.6	16.5	30.2	23.8	23.7
30.5	30.1	21.3	27.1	19.0	25.4	36.5	22.6	25.5	30.0	34.4	30.6	32.7	30.4
29.2	30.2	20.8	30.3	27.8	32.9	28.2	20.6	33.6	22.2	37.5	30.0	24.2	18.8
26.1	29.7	32.0	22.2	29.2	21.5	31.4	43.1	35.9	14.9	24.6	26.2	33.4	29.9
38.8	21.9	25.6	29.7	29.9	32.5	30.4	29.2	40.9	14.1	22.1	20.0	24.3	28.6
22.4	19.9	34.8	33.4	28.0	29.1	27.2	18.8	36.2	27.8	20.2	21.9	27.0	21.9
29.9	21.8	33.1	30.4	30.8	33.9	27.1	27.6	37.2	30.9	31.4	41.9	24.7	25.8
28.3	34.3	34.0	29.0	30.9	24.4	29.0	25.4	30.5	31.1	33.9	15.7	40.5	29.9
26.3	38.3	24.8	23.5	29.3	37.8	29.9	28.6	27.4	29.9	33.5	17.0	34.1	30.9
24.3	20.7	22.5	39.5	32.0	27.6	36.3	22.0	28.4	19.1	25.8	25.7	33.9	43.0
16.8	43.9	27.9	44.4	29.7	23.0	26.8	43.4	29.4	26.7	16.5	22.1	23.0	39.4
27.8	33.1	34.9	20.5	25.4	10.0	28.2	31.0	10.6	28.4	16.5	22.3	17.6	24.2
21.9	27.0	26.5	29.2	24.9	18.4	24.1	28.3	29.0	29.1	18.8	36.7	24.7	23.2
26.2	32.6	22.3	31.7	37.1	35.6	19.5	26.9	24.8	19.2	25.1	22.1	37.9	28.3
29.9	42.1	36.6	25.5	34.2	22.4	40.5	21.2	32.3	31.5	34.2	34.5	39.2	29.3
29.3	31.6	23.2	32.1	20.3	27.8	22.9	32.5	24.5	36.5				

Figure 8.1: Comprimentos (cm) de 500 brotos do arbusto *Banksia ericifolia* de charnecas na Baía de Jervis, Austrália.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

8.3 Importando a planilha



ATENÇÃO

Os links para baixar as planilhas necessárias para repetir esse tutorial podem ser encontrados na seção Arquivos disponíveis do Capítulo Bases de dados.

Ou, baixe aqui o arquivo brotos.xlsx

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações. Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

```
library(openxlsx)
brotos <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/brotos.xlsx",
                     rowNames = F, colNames = F,
                     sheet = "Planilha1")
head(brotos, 10)
head(brotos[, 1:5], 10)
```

```
##      X1     X2     X3     X4     X5     X6     X7     X8     X9     X10
## 1 28.4 29.1 25.7 26.2 25.8 30.2 26.2 35.8 33.4 29.0
## 2 25.1 36.8 29.9 38.3 27.1 32.3 29.0 27.7 28.1 28.9
## 3 29.9 36.3 26.0 21.2 19.8 36.7 21.1 34.6 29.6 32.6
## 4 25.8 23.2 28.8 38.2 32.7 38.7 33.2 24.5 21.6 28.6
## 5 35.0 25.1 24.7 29.5 24.9 29.2 19.5 20.1 30.3 38.9
## 6 36.4 31.8 31.0 39.4 28.8 31.8 28.7 37.0 25.5 19.3
## 7 29.1 21.1 30.4 31.2 38.0 39.0 19.3 27.6 19.1 32.5
## 8 33.2 26.5 38.1 14.9 33.2 27.8 24.7 24.9 25.0 33.1
## 9 26.9 22.5 25.5 33.0 19.4 26.8 24.6 37.5 19.8 43.7
## 10 34.2 33.6 42.5 19.0 25.8 34.0 34.4 42.0 35.4 31.5
```

```
##      X1     X2     X3     X4     X5
## 1 28.4 29.1 25.7 26.2 25.8
## 2 25.1 36.8 29.9 38.3 27.1
## 3 29.9 36.3 26.0 21.2 19.8
## 4 25.8 23.2 28.8 38.2 32.7
## 5 35.0 25.1 24.7 29.5 24.9
## 6 36.4 31.8 31.0 39.4 28.8
## 7 29.1 21.1 30.4 31.2 38.0
## 8 33.2 26.5 38.1 14.9 33.2
## 9 26.9 22.5 25.5 33.0 19.4
## 10 34.2 33.6 42.5 19.0 25.8
```

Exibindo os dados importados (esses comando são “case-sensitive” `ignore.case(object)`).

```
#View(brotos)
print(brotos[1:5,1:5])
brotos
str(brotos)
mode(brotos)
class(brotos)
```

Precisaremos converter para um vetor.

```
brotos_v <- c(t(brotos))
brotos_v
```

```
## [1] 28.4 29.1 25.7 26.2 25.8 30.2 26.2 35.8 33.4 29.0 25.1 36.8 29.9 38.3 27.1
## [16] 32.3 29.0 27.7 28.1 28.9 29.9 36.3 26.0 21.2 19.8 36.7 21.1 34.6 29.6 32.6
## [31] 25.8 23.2 28.8 38.2 32.7 38.7 33.2 24.5 21.6 28.6 35.0 25.1 24.7 29.5 24.9
## [46] 29.2 19.5 20.1 30.3 38.9 36.4 31.8 31.0 39.4 28.8 31.8 28.7 37.0 25.5 19.3
## [61] 29.1 21.1 30.4 31.2 38.0 39.0 19.3 27.6 19.1 32.5 33.2 26.5 38.1 14.9 33.2
## [76] 27.8 24.7 24.9 25.0 33.1 26.9 22.5 25.5 33.0 19.4 26.8 24.6 37.5 19.8 43.7
## [91] 34.2 33.6 42.5 19.0 25.8 34.0 34.4 42.0 35.4 31.5 32.0 38.4 29.1 29.4 29.3
## [106] 26.8 32.4 25.2 28.5 29.8 31.7 22.4 21.7 20.1 21.6 23.5 33.2 33.0 29.6 36.9
## [121] 23.6 16.2 27.3 33.3 21.6 30.2 22.5 33.0 38.3 29.5 31.1 31.7 31.6 41.7 25.9
## [136] 32.3 35.7 31.6 26.0 26.6 29.1 15.8 22.1 23.2 25.4 28.4 20.2 25.5 26.9 32.7
## [151] 29.8 27.1 36.9 32.7 24.8 18.0 40.2 28.0 26.8 41.9 31.5 27.4 37.2 30.6 32.2
## [166] 34.8 28.2 31.3 34.3 32.0 20.6 27.2 25.0 26.1 34.5 44.9 40.5 32.1 40.4 35.7
## [181] 29.3 24.7 37.0 36.9 34.8 29.8 22.8 32.3 34.8 29.6 24.0 30.5 31.6 28.7 20.8
## [196] 14.6 23.4 26.3 31.9 32.5 32.4 36.4 24.1 33.1 26.3 35.7 26.4 34.7 27.5 39.6
## [211] 30.5 30.1 21.3 27.1 19.0 25.4 36.5 22.6 25.5 30.0 29.2 30.2 20.8 30.3 27.8
## [226] 32.9 28.2 20.6 33.6 22.2 26.1 29.7 32.0 22.2 29.2 21.5 31.4 43.1 35.9 14.9
## [241] 38.8 21.9 25.6 29.7 29.9 32.5 30.4 29.2 40.9 14.1 22.4 19.9 34.8 33.4 28.0
## [256] 29.1 27.2 18.8 36.2 27.8 29.9 21.8 33.1 30.4 30.8 33.9 27.1 27.6 37.2 30.9
## [271] 28.3 34.3 34.0 29.0 30.9 24.4 29.0 25.4 30.5 31.1 26.3 38.3 24.8 23.5 29.3
## [286] 37.8 29.9 28.6 27.4 29.9 24.3 20.7 22.5 39.5 32.0 27.6 36.3 22.0 28.4 19.1
```

```
## [301] 16.8 43.9 27.9 44.4 29.7 23.0 26.8 43.4 29.4 26.7 27.8 33.1 34.9 20.5 25.4
## [316] 10.0 28.2 31.0 10.6 28.4 21.9 27.0 26.5 29.2 24.9 18.4 24.1 28.3 29.0 29.1
## [331] 26.2 32.6 22.3 31.7 37.1 35.6 19.5 26.9 24.8 19.2 29.9 42.1 36.6 25.5 34.2
## [346] 22.4 40.5 21.2 32.3 31.5 29.3 34.8 34.8 22.6 33.6 23.0 33.8 21.9 31.6 16.5
## [361] 31.6 26.8 37.3 23.2 20.5 17.6 25.0 41.9 21.8 34.4 23.2 27.6 34.3 26.7 28.1
## [376] 24.7 34.2 15.7 27.2 37.5 32.1 25.5 34.7 43.7 31.6 37.9 27.4 17.0 19.4 24.6
## [391] 20.3 22.4 37.7 29.4 36.0 39.2 26.2 25.7 28.2 22.1 27.8 36.5 23.7 28.6 23.8
## [406] 16.2 38.0 22.1 44.0 20.2 22.9 41.5 30.4 36.1 32.7 30.1 39.9 22.3 26.8 31.4
## [421] 32.5 31.4 18.8 19.1 24.2 29.3 19.7 36.7 24.1 33.9 24.5 22.8 29.9 34.5 33.4
## [436] 22.8 30.8 22.1 38.1 33.5 36.5 33.5 28.6 24.9 24.3 25.7 19.2 34.5 40.6 25.8
## [451] 39.4 33.3 21.9 29.6 27.0 30.2 17.1 15.8 22.8 16.5 24.2 21.2 25.8 29.8 24.7
## [466] 30.6 38.1 33.3 26.8 16.5 23.2 24.5 29.9 26.0 40.5 30.0 30.3 33.9 34.9 18.8
## [481] 28.3 40.2 30.9 22.0 34.1 26.2 19.6 33.6 26.3 25.1 29.3 31.9 43.0 31.9 33.9
## [496] 20.0 39.6 34.3 28.8 34.2
```

E agora visualizando nossos dados.

```
#View(brotos_v)
print(brotos_v)
brotos_v
str(brotos_v)
mode(brotos_v)
class(brotos_v)
```

Tendo em mãos o conjunto total de 500 comprimentos medidos de brotos de *Banksia*, agora **vamos tirar uma subamostra aleatória de uma parte dos 500 valores**. Essa subamostra é tirada usando o comando `size=` no código subsequente, que estabelece o tamanho da subamostra a ser tirada do total de dados. Esse tamanho é estabelecido no quiz. Inclua esse valor de acordo com o que é pedido no quiz. Por exemplo, se for pedido uma subamostra de 150 comprimentos, então `size = 150`. Nesse tutorial usaremos todos os 500 comprimentos.

```
set.seed(666)
brotos_sub <- sample(brotos_v, size = 500, replace = F) #atualize o valor de 'size=' s...
```

```
## [1] 21.1 30.2 34.0 36.0 27.3 34.6 15.8 19.4 31.6 26.6 31.5 38.9 31.1 32.0 29.9
## [16] 24.5 21.2 19.3 22.1 24.7 24.7 37.0 22.4 32.5 35.9 36.9 24.2 33.0 27.6 22.8
## [31] 26.5 31.6 32.3 38.1 27.4 32.3 24.8 25.5 33.1 34.5 24.4 27.1 43.9 28.6 28.4
## [46] 33.4 29.3 24.9 36.2 24.6 30.0 32.6 25.6 26.0 25.8 37.0 26.4 22.5 23.2 38.2
## [61] 23.0 44.4 27.6 20.2 29.0 25.9 22.3 27.4 14.9 33.3 40.9 33.5 36.3 28.0 28.4
## [76] 41.5 28.5 34.8 29.1 31.6 36.7 30.1 35.4 19.2 39.4 16.5 25.1 28.4 31.9 16.2
## [91] 31.7 29.3 33.9 22.3 33.6 25.5 33.0 30.0 33.3 34.0 44.0 29.5 33.6 26.7 42.0
## [106] 33.2 24.8 26.9 25.4 43.0 29.4 15.8 28.8 39.5 31.8 20.3 38.3 19.8 37.8 26.7
## [121] 23.0 19.8 39.6 18.4 31.0 29.1 24.0 32.4 22.6 30.9 26.2 29.1 28.2 33.4 25.7
```

```

## [136] 30.9 34.8 34.1 24.1 29.9 20.0 29.3 28.7 28.6 36.5 25.1 17.1 24.7 29.2 38.7
## [151] 26.8 30.6 17.0 25.5 26.2 31.5 32.5 34.7 30.2 36.6 36.5 29.1 32.3 35.0 33.9
## [166] 19.1 25.0 26.1 40.2 30.4 31.4 33.6 37.5 29.8 20.8 29.3 39.6 28.4 31.0 24.9
## [181] 27.5 33.4 33.3 34.3 21.6 40.5 29.4 26.3 22.5 42.5 20.5 44.9 29.8 30.4 28.6
## [196] 27.6 39.4 33.2 34.2 20.1 18.8 26.8 22.9 25.7 21.9 34.9 34.8 33.2 40.6 26.9
## [211] 20.2 19.1 27.2 23.2 37.3 32.5 25.4 32.0 34.5 29.7 28.1 31.2 27.2 25.8 27.0
## [226] 26.3 32.2 43.7 42.1 38.4 19.0 28.9 31.8 34.5 22.1 29.0 25.2 38.1 29.2 34.3
## [241] 27.9 24.5 34.4 40.2 36.1 29.6 33.1 28.8 22.8 38.0 20.1 28.6 34.4 30.5 19.5
## [256] 30.2 30.4 22.4 35.6 17.6 28.2 29.5 37.7 19.5 28.0 37.1 20.6 34.3 40.5 27.8
## [271] 29.9 27.1 36.9 25.0 29.9 32.7 33.2 24.7 30.9 27.0 29.9 29.6 29.7 23.8 22.2
## [286] 34.8 38.8 31.1 26.8 19.0 22.4 36.9 35.7 22.0 24.1 26.1 34.3 25.5 18.0 41.7
## [301] 29.8 32.0 31.4 21.8 27.7 33.1 28.7 29.2 31.6 23.6 34.2 25.8 21.3 26.3 23.5
## [316] 21.6 32.7 27.8 32.7 30.3 16.8 33.0 10.0 30.4 25.4 26.2 22.1 37.5 26.0 33.5
## [331] 29.0 36.7 29.8 24.8 31.9 32.6 36.4 32.1 31.5 33.8 43.4 29.4 24.2 25.0 34.9
## [346] 22.0 24.9 16.2 28.3 16.5 29.1 24.6 19.3 25.8 29.0 29.3 29.9 32.7 35.8 32.3
## [361] 19.9 30.2 21.2 24.3 32.5 27.6 25.8 39.9 34.7 25.5 29.3 21.9 23.2 26.2 19.4
## [376] 19.2 32.1 26.3 26.8 38.0 28.3 34.8 19.6 35.7 35.7 23.2 32.9 27.4 22.8 40.4
## [391] 37.9 28.2 43.7 21.7 21.1 31.6 24.9 25.5 29.6 40.5 39.2 39.0 29.1 33.9 25.7
## [406] 29.6 29.7 19.7 38.3 22.6 36.4 16.5 28.3 30.8 21.2 14.9 34.8 33.6 24.1 25.4
## [421] 43.1 20.7 30.3 41.9 31.3 27.1 34.2 27.8 18.8 31.7 30.6 27.8 21.6 30.1 30.8
## [436] 27.2 28.8 37.2 14.6 18.8 26.9 22.1 14.1 25.1 22.2 36.5 23.4 29.9 33.9 24.3
## [451] 33.1 26.2 28.1 24.5 34.2 15.7 32.0 22.8 21.9 26.0 23.2 20.8 21.5 38.1 30.3
## [466] 19.1 31.6 27.1 32.4 23.5 36.8 37.2 21.8 26.8 31.9 22.4 24.7 20.6 30.5 26.8
## [481] 30.5 29.0 26.5 29.2 31.4 20.5 26.8 10.6 27.8 21.9 28.2 23.7 29.9 29.9 36.3
## [496] 31.7 38.3 29.2 22.5 41.9

```

Para fazermos uma tabela de frequência do comprimento dos brotos carregamos o pacote `fdth` e pedimos a função `range` que retorna o valor máximo e mínimo no vetor.

```

library(fdth)
range(brotos_sub) #retorna o valor máximo e mínimo
#?range

```

```

## [1] 10.0 44.9

```

Agora é necessário que você substitua os valores de range 10, 44.9 nos valores de início e fim da distribuição de frequência. Os comandos abaixo criam uma tabela de frequência chamada `tf` com valores máximos e mínimos definidos por `range(brotos_v)` em intervalo definidos por `h=5`, e o comando `print(tf)` exibe a tabela de frequência.

```

tf <- fdt(brotos_sub, start=10, end=45, h=5) #tabela de frequência
#?fdt #atente para o uso de k, por exemplo tf <- fdt(brotosv, k=5)
print(tf)

```

```

##   Class limits   f   rf rf(%)   cf cf(%)
##   [10,15)    6  0.01   1.2    6   1.2
##   [15,20)   35  0.07   7.0   41   8.2
##   [20,25)   93  0.19  18.6  134  26.8
##   [25,30)  155  0.31  31.0  289  57.8
##   [30,35)  130  0.26  26.0  419  83.8
##   [35,40)   57  0.11  11.4  476  95.2
##   [40,45)   24  0.05   4.8  500 100.0

```

Class limits = limites de classe, f = frequência de classe, rf = frequência relativa da classe, rf(%) = frequência percentual da classe, cf = frequência cumulativa da classe, cf(%) = frequência cumulativa percentual da classe.

Se contarmos o número de brotos que se encontram em cada um dos sete intervalos, temos a base para a tabulação da frequência. Essa tabulação é mostrada na tabela gerada pelos códigos acima. A coluna de frequência foi obtida contando o número de medições que existem dentro de cada classe. A coluna de frequência percentual foi obtida representando cada contagem como uma porcentagem da contagem total. A frequência cumulativa e as frequências percentuais cumulativas foram obtidas somando progressivamente as frequências correspondentes.

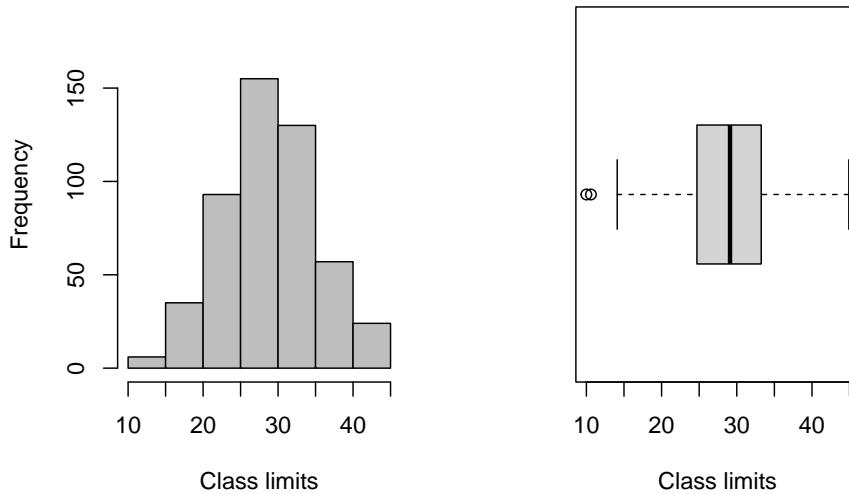
8.4 Gráficos de histograma e boxplot

No código abaixo se define o layout dos gráficos para serem exibidos lado a lado, e na sequência criamos um gráfico de histograma com base na tabela de frequência tf. O comando `boxplot()` cria um gráfico de boxplot para os valores em `brotos_v`. O argumento “horizontal = TRUE” indica que o boxplot deve ser horizontal.

```

par(mfrow = c(1,2)) #gráficos lado a lado
plot(tf) #distribuição de frequências
boxplot(brotos_sub, horizontal = TRUE,
        xlab="Class limits") #boxplot
par(mfrow = c(1,1)) #gráficos de volta ao normal

```



Se você recebeu a mensagem de erro “Error in plot.new() : figure margins too large”, aumente o tamanho da janela do gráfico e execute as últimas três linhas de comando novamente.

```
summary(brotos_v)
##summary
sd(brotos_v) #desvio padrão
var(brotos_v) #variância
```

```
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 10.00   24.70  29.10   28.97  33.23   44.90
## [1] 6.373581
## [1] 40.62254
```

Apêndices

Sites consultados

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executa-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

```
## rm(list=ls(all=TRUE)) #limpa a memória
## install.packages("openxlsx") #importa arquivos do excel
## install.packages("fdth")
library(openxlsx)
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
library(openxlsx)
brotos <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/brotos.xlsx",
                     rowNames = F, colNames = F,
                     sheet = "Planilha1")
head(brotos,10)
head(brotos[, 1:5], 10)
## #View(brotos)
## print(brotos[1:5,1:5])
## brotos
## str(brotos)
## mode(brotos)
## class(brotos)
brotos_v <- c(t(brotos))
brotos_v
## #View(brotosv)
## print(brotos_v)
## brotos_v
## str(brotos_v)
## mode(brotos_v)
## class(brotos_v)
set.seed(666)
brotos_sub <- sample(brotos_v, size = 500, replace = F) #atualize o valor de 'size=' se quiser
brotos_sub
library(fdth)
range(brotos_sub) #retorna o valor máximo e mínimo
##?range
tf <- fdt(brotos_sub, start=10, end=45, h=5) #tabela de frequência
##?fdt #atente para o uso de k, por exemplo tf <- fdt(brotosv, k=5)
print(tf)
```

```
par(mfrow = c(1,2)) #gráficos lado a lado
plot(tf) #distribuição de frequências
boxplot(brotos_sub, horizontal = TRUE,
        xlab="Class limits") #boxplot
par(mfrow = c(1,1)) #gráficos de volta ao normal
summary(brotos_v)
#?summary
sd(brotos_v) #desvio padrão
var(brotos_v) #variância
```

Referências

Chapter 9

R Modulo 3.2 - Testes estatísticos

RESUMO

A estatística descritiva tem um papel importante a desempenhar na ciência. Quando problemas específicos são tratados na ciência, os dados precisam ser coletados, analisados e apresentados de forma concisa para que outros possam se beneficiar do que foi encontrado.

Apresentação

A estatística descritiva tem um papel importante a desempenhar na ciência. Quando problemas específicos são tratados na ciência, os dados precisam ser coletados, analisados e apresentados de forma concisa para que outros possam se beneficiar do que foi encontrado. Geralmente não é possível apresentar um conjunto de dados completo em uma publicação ou em um seminário e, mesmo que fosse, é improvável isso permitisse uma boa comunicação dos resultados da pesquisa. Em vez disso, os dados são geralmente resumidos como tabelas de frequência, histogramas e estatísticas descritivas que os leitores ou ouvintes podem assimilar prontamente, mas que ainda transmitem os elementos essenciais do conjunto de dados original. O principal objetivo do cálculo das estatísticas descritivas é transmitir informações essenciais contidas em um conjunto de dados da forma mais concisa e clara possível.

9.1 Sobre os dados

Considere os dados merísticos (ou medições morfológicas) da espécie de peixe *Cichla ocellaris* (tucunaré amarelo) do reservatório da barragem de Gramame,

PB (MEDEIROS; ROSA, 1994) (Figura 10.1). Existem 421 medições do comprimento total (CT), comprimento padrão (CP) e peso total (PT), além do sexo (MACHO, FÊMEA ou imaturo), e outros descritores da estrutura populacional da espécie, um conjunto de dados formidável.

9.2 Organização básica

```
dev.off() #apaga os graficos, se houver algum
rm(list=ls(all=TRUE)) #limpa a memoria
cat("\014") #limpa o console
```

Instalando os pacotes necessários para esse módulo

```
install.packages("openxlsx") #importa arquivos do excel
install.packages("fdth")
install.packages("ggpubr")
install.packages("multcomp")
```

```
library(openxlsx)
```

Os códigos acima, são usados para instalar e carregar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades do R (R CRAN (TEAM, 2017) e RStudio (TEAM, 2022)). No exemplo acima, o pacote `openxlsx` permite ler e escrever arquivos Excel no R. Para instalar um pacote no R, você precisa usar a função `install.packages()`.

Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`. Por exemplo, para carregar o pacote `openxlsx`, você precisa executar a função `library(openxlsx)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

n = 421	mes	periodo	estação	sexo	CT_cm	CP_cm	PT_g
TU 0001	ago	chuvisco	inverno	MACHO	32,4	27,2	468,8
TU 0002	ago	chuvisco	inverno	MACHO	33,4	28,8	520,0
TU 0003	ago	chuvisco	inverno	MACHO	27,3	23,8	301,5
TU 0004	ago	chuvisco	inverno	MACHO	13,2	11,0	28,2
TU 0005	ago	chuvisco	inverno	MACHO	14,3	11,9	38,9
TU 0006	set	chuvisco	inverno	MACHO	22,7	20,5	431,7
TU 0007	set	chuvisco	inverno	FEMEA	23,2	19,2	544,0
TU 0008	set	chuvisco	inverno	imaturo	13,5	11,5	161,6
TU 0009	set	chuvisco	inverno	MACHO	24,6	20,5	200,5
TU 0010	set	chuvisco	inverno	MACHO	19,4	16,0	86,7
TU 0011	set	chuvisco	inverno	FEMEA	25,5	21,2	206,7
TU 0012	set	chuvisco	inverno	MACHO	23,0	18,9	153,0
TU 0013	set	chuvisco	inverno	MACHO	17,4	14,4	68,7
TU 0014	set	chuvisco	inverno	FEMEA	16,5	12,2	53,6
TU 0015	set	chuvisco	inverno	MACHO	21,9	18,4	141,1
TU 0016	set	chuvisco	inverno	MACHO	28,1	23,4	299,3
TU 0017	set	chuvisco	inverno	MACHO	27,2	22,2	253,6
TU 0018	set	chuvisco	inverno	MACHO	25,5	20,6	226,0
TU 0019	set	chuvisco	inverno	FEMEA	38,0	32,0	437,1
TU 0020	set	chuvisco	inverno	MACHO	27,5	22,8	299,8
TU 0021	set	chuvisco	inverno	imaturo	7,8	6,2	4,5
TU 0022	set	chuvisco	inverno	imaturo	6,6	5,1	2,1
TU 0023	set	chuvisco	inverno	imaturo	7,8	6,1	4,0
TU 0024	set	chuvisco	inverno	imaturo	8,3	6,6	5,5
TU 0025	set	chuvisco	inverno	imaturo	8,6	6,8	6,7
TU 0026	set	chuvisco	inverno	imaturo	6,3	5,9	2,5
TU 0027	set	chuvisco	inverno	imaturo	8,0	6,1	4,9
TU 0028	set	chuvisco	inverno	imaturo	10,9	8,5	13,5
TU 0029	set	chuvisco	inverno	imaturo	7,1	5,5	3,3
TU 0030	set	chuvisco	inverno	imaturo	9,8	7,8	10,0
TU 0031	set	chuvisco	inverno	imaturo	6,0	4,6	2,1
TU 0032	set	chuvisco	inverno	imaturo	6,1	5,9	2,0
TU 0033	set	chuvisco	inverno	imaturo	6,2	4,9	1,9
TU 0034	set	chuvisco	inverno	imaturo	6,6	5,2	2,3
TU 0035	set	chuvisco	inverno	imaturo	6,5	5,0	2,5
TU 0036	set	chuvisco	inverno	imaturo	6,1	4,8	2,2
TU 0037	set	chuvisco	inverno	imaturo	6,1	4,8	2,5
<hr/>							

Figure 9.1: Dados merísticos da espécie de peixe **Cichla ocellaris** (tucunaré amarelo) do reservatório da barragem de Gramame, PB.

9.3 Importando a planilha



ATENÇÃO

Os links para baixar as planilhas necessárias para repetir esse tutorial podem ser encontrados na seção Arquivos disponíveis do Capítulo Bases de dados.

Ou, baixe aqui o arquivo tucuna.xlsx

Vamos importar a planilha de dados univariados `tucuna.xlsx`. Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações. Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

```
library(openxlsx)
univ <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/tucuna.xlsx",
                   rowNames = F, colNames = T,
                   sheet = "tucuna")
head(univ, 10)
head(univ[, 1:5], 10)

##      n.=.434 CT_cm  PT_g CP_cm Ctubo_cm PC_g      %PT Pest_g Cest_cm gr_est
## 1    TU001 32.4 468.8 27.2   39.8 458.9 2.111775 3.9     7.7     I
## 2    TU002 33.4 520.0 28.8   14.3 507.4 2.423077 5.9    10.0     I
## 3    TU003 27.3 301.5 23.8   13.0 283.4 6.003317 15.5   10.2    III
## 4    TU004 13.2 28.2 11.0   16.5 27.7 1.773050 0.3     4.3     I
## 5    TU005 14.3 38.9 11.9   15.5 37.7 3.084833 0.8     4.5    III
## 6    TU006 22.7 431.7 20.5   24.2 418.7 3.011350 9.9     6.4    III
## 7    TU007 23.2 544.0 19.2   25.0 520.1 4.393382 6.8     7.5    II
## 8    TU008 13.5 161.6 11.5   17.5 157.0 2.846535 4.1     6.8    II
## 9    TU009 24.6 200.5 20.5   25.7 195.9 2.294264 2.5     7.6    II
## 10   TU010 19.4  86.7 16.0   18.3  84.5 2.537486 0.7     5.1     I
##          ir_est Pint_g Cint_cm gr_int   ir_int Pgon_g Cgon_cm      emg
## 1  0.8319113    4.8   38.3    II 1.0238908    1.2      7  IMATURO
## 2  1.1346154    5.3   13.3    II 1.0192308    1.4     6.5  MADURO
## 3  5.1409619    2.4   12.0    II 0.7960199    0.2     7.7 EM MATERACAO
## 4  1.0638298    0.1   16.0    II 0.3546099    0.1      5  IMATURO
## 5  2.0565553    0.3   15.0    II 0.7712082    0.1     3.2  IMATURO
## 6  2.2932592    2.7   23.7    II 0.6254343    0.4     7.8 EM MATERACAO
## 7  1.2500000    3.3   24.5    II 0.6066176   13.8     7.9  MADURO
## 8  2.5371287    0.4   17.0    I  0.2475248    0.1     2.5  IMATURO
## 9  1.2468828    2.0   25.3    II 0.9975062    0.1     6.5 EM MATERACAO
## 10 0.8073818    1.4   17.7    II 1.6147636    0.1      6  IMATURO
```

```

##          ig mes periodo estação   sexo
## 1  0.25597270 ago chuvoso inverno MACHO
## 2  0.26923077 ago chuvoso inverno MACHO
## 3  0.06633499 ago chuvoso inverno MACHO
## 4  0.35460993 ago chuvoso inverno MACHO
## 5  0.25706941 ago chuvoso inverno MACHO
## 6  0.09265694 set chuvoso inverno MACHO
## 7  2.53676471 set chuvoso inverno FEMEA
## 8  0.06188119 set chuvoso inverno imaturo
## 9  0.04987531 set chuvoso inverno MACHO
## 10 0.11534025 set chuvoso inverno MACHO
##    n.=.434 CT_cm  PT_g CP_cm Ctubo_cm
## 1    TU001  32.4 468.8  27.2     39.8
## 2    TU002  33.4 520.0  28.8     14.3
## 3    TU003  27.3 301.5  23.8     13.0
## 4    TU004  13.2  28.2  11.0     16.5
## 5    TU005  14.3  38.9  11.9     15.5
## 6    TU006  22.7 431.7  20.5     24.2
## 7    TU007  23.2 544.0  19.2     25.0
## 8    TU008  13.5 161.6  11.5     17.5
## 9    TU009  24.6 200.5  20.5     25.7
## 10   TU010  19.4  86.7  16.0     18.3

```

Exibindo os dados importados (esses comando são “case-sensitive” `ignore.case(object)`).

```

#View(univ)
print(univ[1:5,1:5])
univ
str(univ)
mode(univ)
class(univ)

```

Vamos escolher uma coluna como a variável de interesse para trabalhar com ela. No código abaixo, essa coluna é descrita pelo seu nome apresentado depois do \$. Antes do \$ especificamos e qual data frame está a variável. Depois disso, a convertemos para um vetor.

```

var <- univ$CP_cm
var_v <- as.vector(var)
range(var_v)

```

```
## [1]  3.5 36.1
```

E agora visualizando nossos dados.

```
#View(var)
print(var_v)
var_v
str(var_v)
mode(var_v)
class(var_v)
range(var_v)
```

Por inspeção, o comprimento total mínimo do peixe é 3.5 cm e o máximo é 36.1 cm. Esses valores definem o intervalo da amostra. Agora precisamos subdividir os dados em intervalos ou classes, cada um com o mesmo tamanho. Geralmente, é aconselhável arredondar o valor mínimo para baixo e o valor máximo para cima, para valores apropriados ao decidir as classes de intervalos. Nesse caso, parece sensato dividir a faixa de `range(var_v)` cm em sete intervalos a cada 5 cm de largura. Se contarmos o número de peixes que se encontram em cada um dos sete intervalos, temos a base para a tabulação da frequência.

A coluna de frequência será obtida contando o número de medições que existem dentro de cada classe. A coluna de frequência percentual será obtida representando cada contagem como uma porcentagem da contagem total. A frequência cumulativa e as frequências percentuais cumulativas serão obtidas somando progressivamente as frequências correspondentes.

Tendo em mãos o conjunto total de 434 valores merísticos da variável de interesse agora **podemos tirar uma subamostra aleatória de uma parte dos 434 valores**. Essa subamostra é tirada usando o comando `size=` no código subsequente, que estabelece o tamanho da subamostra a ser tirada do total de dados. Esse tamanho é estabelecido no quiz. Inclua esse valor de acordo com o que é pedido no quiz. Por exemplo, se for pedido uma subamostra de 150 comprimentos, então `size = 150`. Nesse tutorial usaremos todos os 434 comprimentos. Podemos ainda escolher uma das colunas da base de dados, no caso desse vamos usar a coluna

Substitua em `n <-` o valor de `size=` desejado. Aqui `n <-` será todo o conjunto de dados `length(var_v)`.

```
n <- length(var_v)
set.seed(666)
var_sub <- sample(var_v, size = n, replace = F) #atualize o valor de 'size=' se necessário
#OU
#var_sub <- univ[univ$CP_cm >0 & univ$CP_cm <20,] #data.frame
#var_sub <- var_v[var_v>=5 & var_v <=15] #vector
```

Para fazermos uma tabela de frequência dos valores merísticos carregamos o pacote `fdth` e pedimos a função `range` que retorna o valor máximo e mínimo no vetor.

```
library(fdth)
range(var_sub) #retorna o valor máximo e mínimo
#?range
```

```
## [1] 3.5 36.1
```

Agora é necessário que você substitua os valores de range 3.5, 36.1 nos valores de início e fim da distribuição de frequência. Os comandos abaixo criam uma tabela de frequência chamada `tf` com valores máximos e mínimos definidos por `range(var_v)` em intervalo definidos por `h=5`, e o comando `print(tf)` exibe a tabela de frequência.

```
tf <- fdt(var_sub, start=0, end=40, h=5) #tabela de frequência
#?fdt #atente para o uso de k, por exemplo tf <- fdt(var_sub, k=5)
print(tf)
```

	Class limits	f	rf	rf(%)	cf	cf(%)
##	[0,5)	57	0.13	13.13	57	13.13
##	[5,10)	154	0.35	35.48	211	48.62
##	[10,15)	109	0.25	25.12	320	73.73
##	[15,20)	38	0.09	8.76	358	82.49
##	[20,25)	33	0.08	7.60	391	90.09
##	[25,30)	23	0.05	5.30	414	95.39
##	[30,35)	15	0.03	3.46	429	98.85
##	[35,40)	5	0.01	1.15	434	100.00

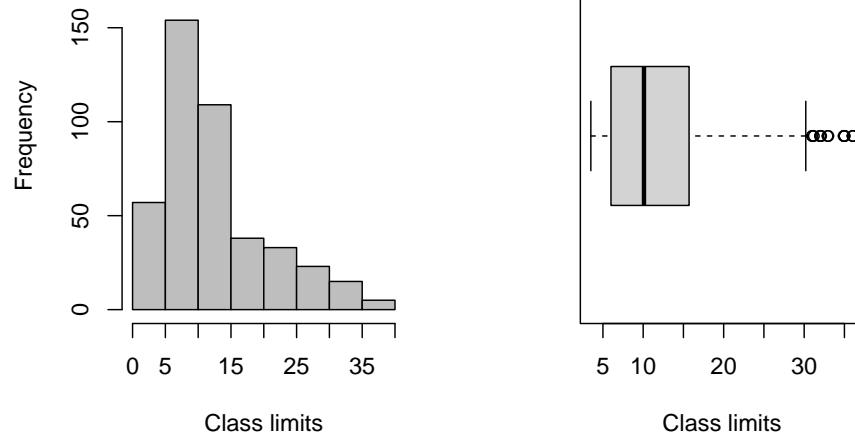
Class limits = limites de classe, f = frequência de classe, rf = frequência relativa da classe, rf(%) = frequência relativa percentual da classe, cf = frequência cumulativa da classe, cf(%) = frequência cumulativa percentual da classe.

Se contarmos o número de brotos que se encontram em cada um dos sete intervalos, temos a base para a tabulação da frequência. Essa tabulação é mostrada na tabela gerada pelos códigos acima. A coluna de frequência foi obtida contando o número de medições que existem dentro de cada classe. A coluna de frequência percentual foi obtida representando cada contagem como uma porcentagem da contagem total. A frequência cumulativa e as frequências percentuais cumulativas foram obtidas somando progressivamente as frequências correspondentes.

9.4 Gráficos de histograma e boxplot

No código abaixo se define o layout dos gráficos para serem exibidos lado a lado, e na sequência criamos um gráfico de histograma com base na tabela de frequência `tf`. O comando `boxplot()` cria um gráfico de boxplot para os valores em `var_v`. O argumento “horizontal = TRUE” indica que o boxplot deve ser horizontal.

```
par(mfrow = c(1,2)) #gráficos lado a lado
plot(tf) #distribuição de frequências
boxplot(var_sub, horizontal = TRUE,
       xlab="Class limits") #boxplot
```



```
par(mfrow = c(1,1)) #gráficos de volta ao normal
```

IMPORTANTE Se você recebeu a mensagem de erro “Error in plot.new() : figure margins too large”, aumente o tamanho da janela do gráfico e execute as últimas três linhas de comando novamente.

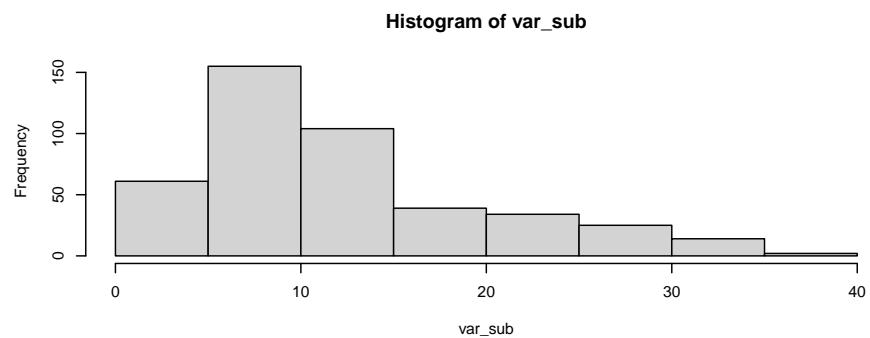
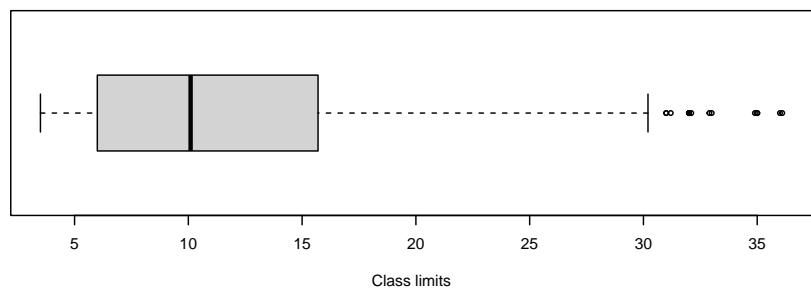
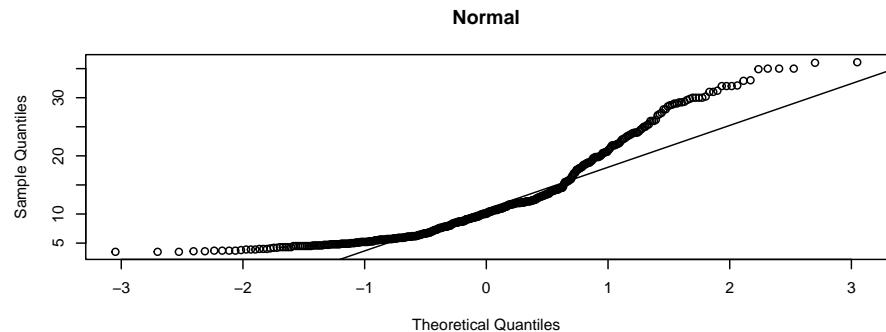
9.5 Sumário estatístico geral

```
summary(var_sub)
#?summary
sd(var_sub) #desvio padrão
var(var_sub) #variância
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	3.50	6.00	10.10	12.32	15.70	36.10
## [1]	7.845708					
## [1]	61.55514					

9.6 Testando normalidade

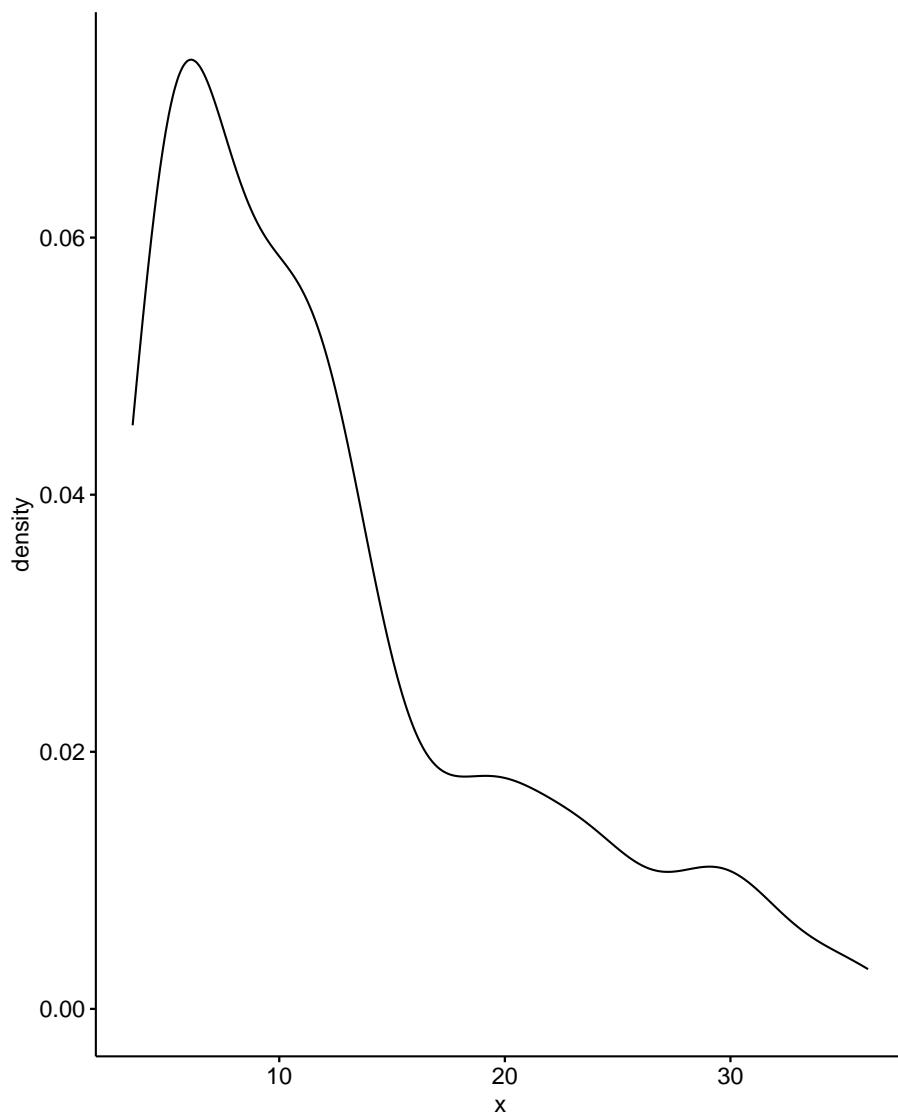
```
par(mfrow=c(3,1))
qqnorm(var_sub, main='Normal')
qqline(var_sub)
boxplot(var_sub, horizontal = TRUE,
       xlab="Class limits") #boxplot
hist(var_sub)
```



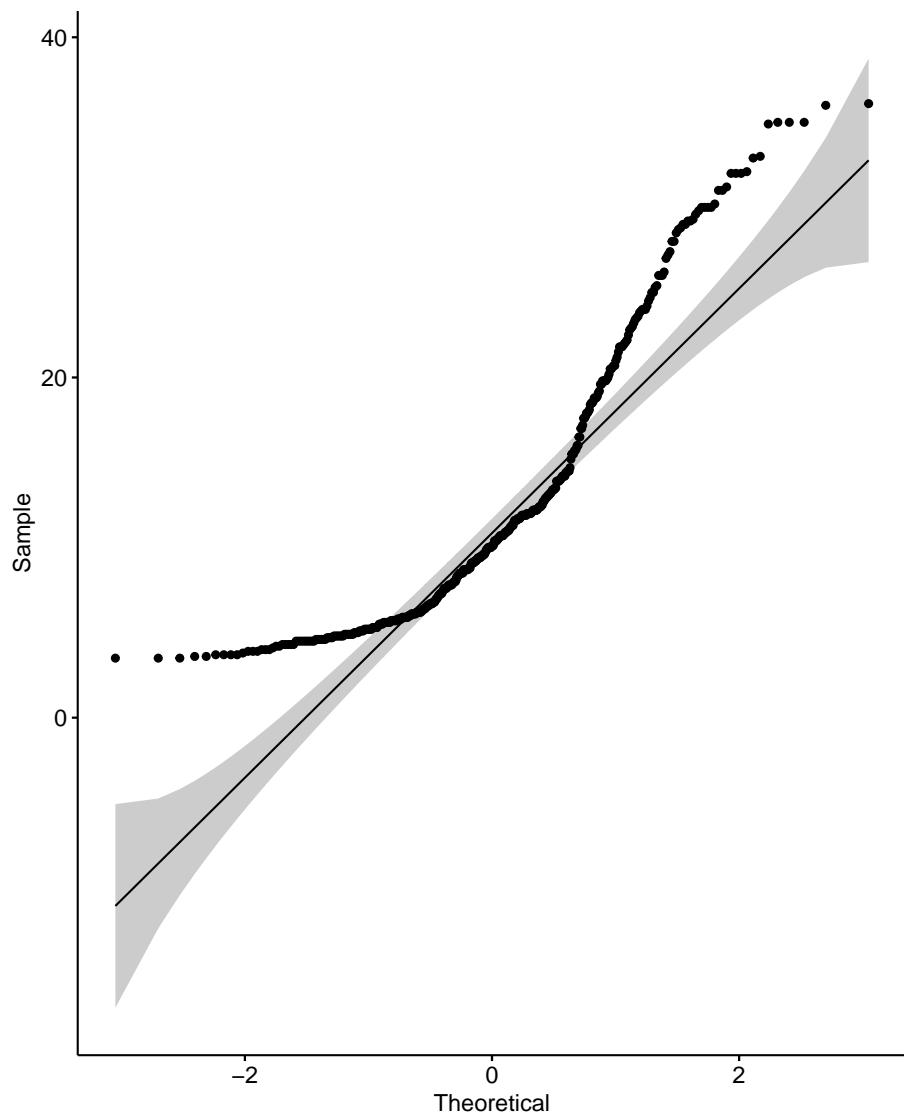
```
par(mfrow=c(1,1))
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.3.3
## Carregando pacotes exigidos: ggplot2
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
ggdensity(var_sub)
```

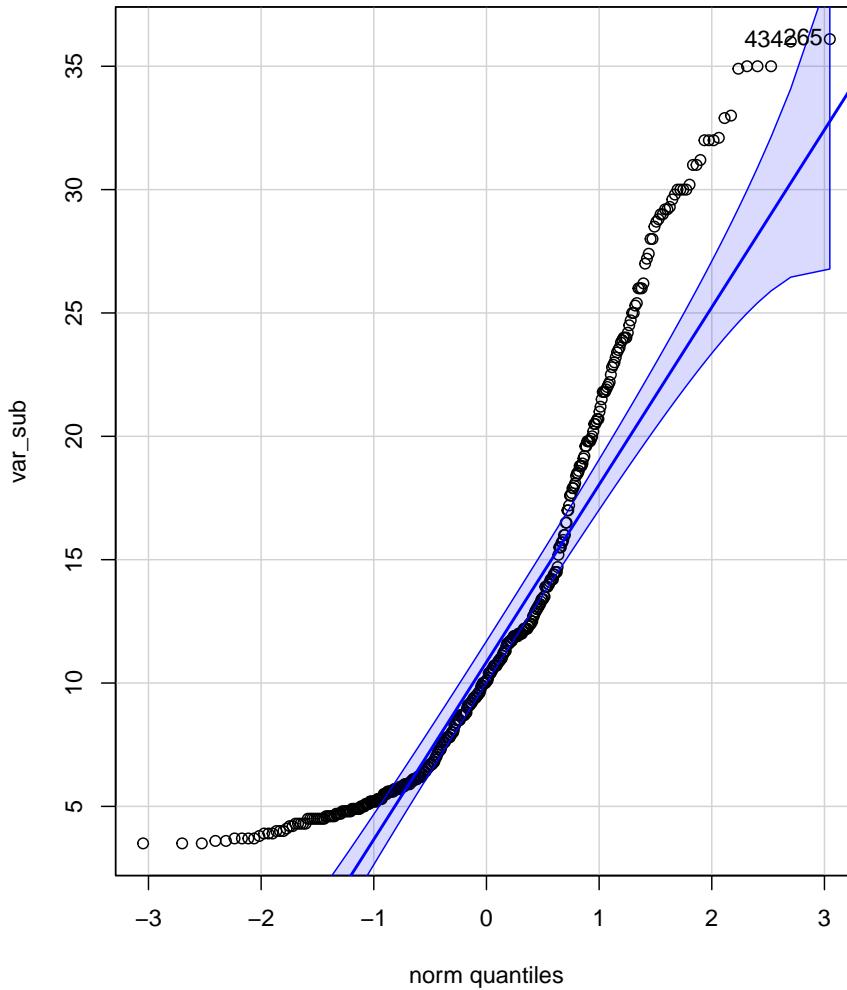


```
ggqqplot(var_sub)
```



```
library(car)
```

```
## Carregando pacotes exigidos: carData
qqPlot(var_sub)
```



```
## [1] 265 434
```

9.6.2 Testes de Shapiro-Wilk e Kolmogorov-Smirnov

```
shap <- shapiro.test(var_sub)
shap
p <- format(shap$p.value, scientific = FALSE)
p
```

```

ks.test(var_sub, "pnorm")

## Warning in ks.test.default(var_sub, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test

##
## Shapiro-Wilk normality test
##
## data: var_sub
## W = 0.86837, p-value < 2.2e-16
##
## [1] "0.0000000000000000107347"
##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: var_sub
## D = 0.99977, p-value < 2.2e-16
## alternative hypothesis: two-sided

```

9.7 Testando diferenças entre médias

9.7.1 Testando homogeneidade de variâncias

```

# F-test
#var.test(CT_cm ~ sexo, data = univ) com erro
#Interpretação: Um valor de p maior que o nível de significância de 0.05 significa que

# Levene
library(car)
#univ$sexo <- as.factor(univ$sexo) #evita o Warning de "group coerced to factor"
lev <- leveneTest(CT_cm ~ sexo, data = univ)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

lev

# Teste de Levene entre dois de tres grupos
machos_CT_cm <- na.omit(univ$CT_cm[univ$sexo == "MACHO"])
femeas_CT_cm <- na.omit(univ$CT_cm[univ$sexo == "FEMEA"])
imat_CT_cm <- na.omit(univ$CT_cm[univ$sexo == "imaturo"])
lev2 <- leveneTest(CT_cm ~ sexo, data = univ[univ$sexo %in% c("MACHO", "FEMEA"), ])

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to

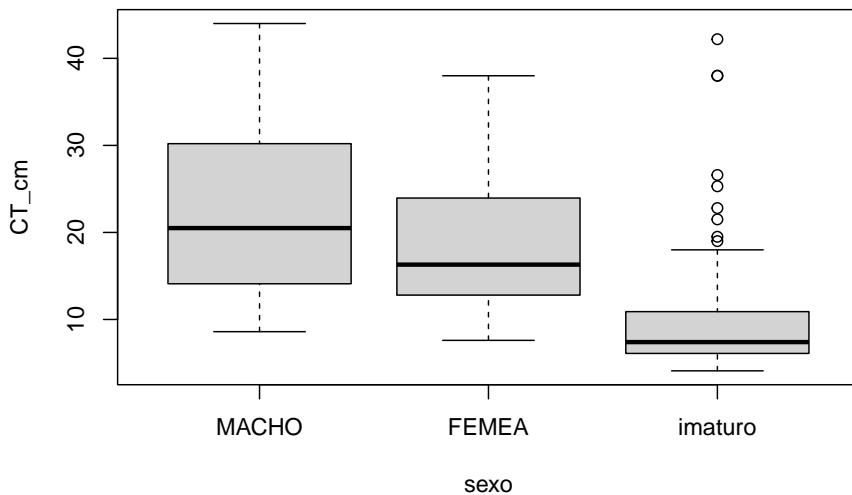
```

```
## factor.
```

```
lev2
```

#Interpretação: Um valor de p maior que o nível de significância de 0.05 significa que, a hipótese nula é aceita.

```
univ$sexo <- factor(univ$sexo, levels = c("MACHO", "FEMEA", "imaturo"))
boxplot(CT_cm ~ sexo, data = univ)
```



```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     2  45.395 < 2.2e-16 ***
##          431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  10.826 0.001173 **
##          211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.7.2 Teste entre duas médias (t-Student)

```
t.test(machos_CT_cm, femeas_CT_cm,
       alternative = c("two.sided", "less", "greater"),
```

```

    mu = 0, paired = FALSE, var.equal = FALSE,
    conf.level = 0.95)

## Welch Two Sample t-test
##
## data: machos_CT_cm and femeas_CT_cm
## t = 3.4342, df = 210.89, p-value = 0.0007155
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 1.747453 6.456834
## sample estimates:
## mean of x mean of y
## 22.70984 18.60769

```

9.7.3 Teste entre três médias (ANOVA)

ANOVA

```

levels(univ$sexo)
univ$sexo <- ordered(univ$sexo,
                      levels = c("MACHO", "FEMEA", "imaturo"))
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

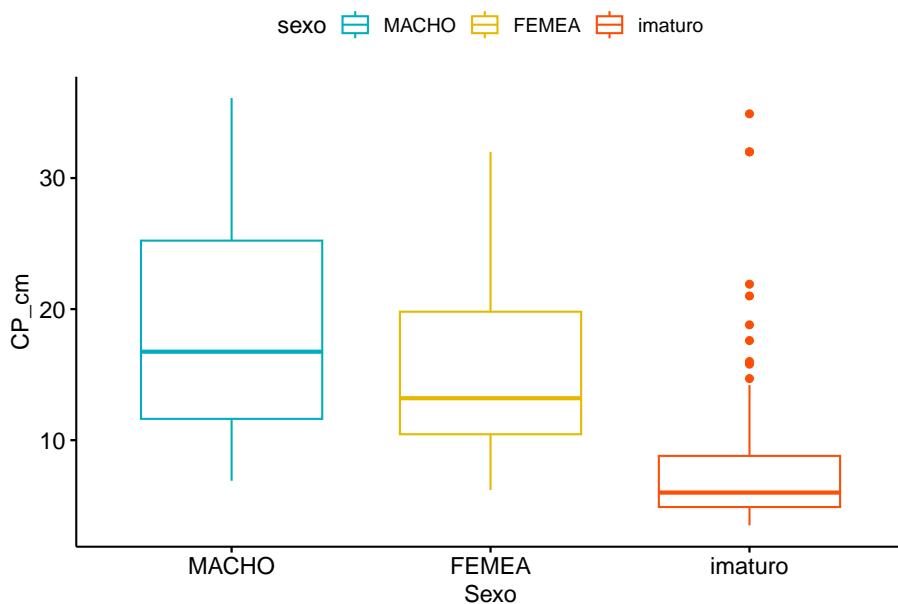
group_by(univ, sexo) %>%
  summarise(
    count = n(),

```

```

mean = mean(CP_cm, na.rm = TRUE),
sd = sd(CP_cm, na.rm = TRUE)
)
# Conjunto de gráficos
library("ggpubr")
ggboxplot(univ, x = "sexo", y = "CP_cm",
          color = "sexo", palette = c("#00AFBB", "#E7B800", "#FC4E07"),
          order = c("MACHO", "FEMEA", "imaturo"),
          ylab = "CP_cm", xlab = "Sexo")

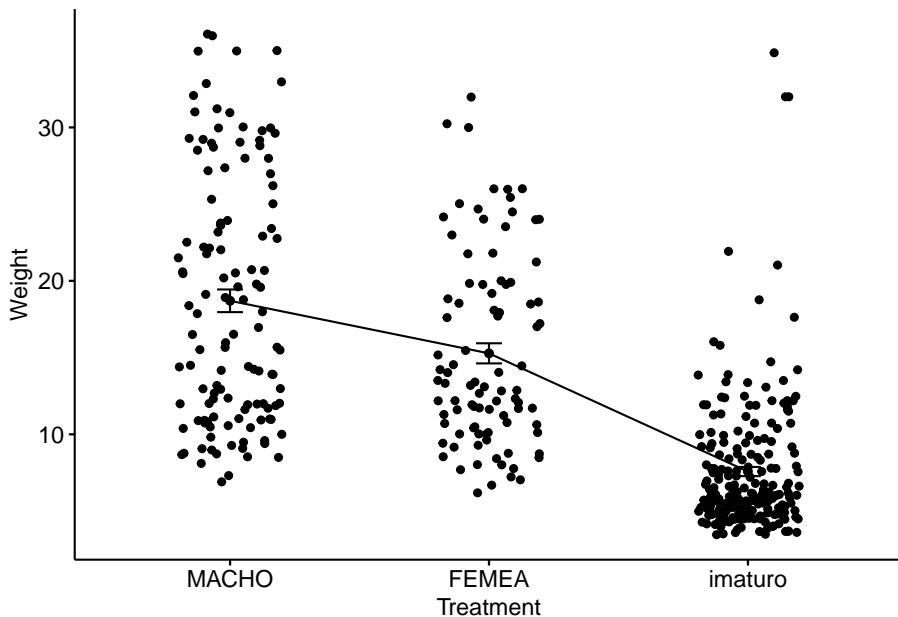
```



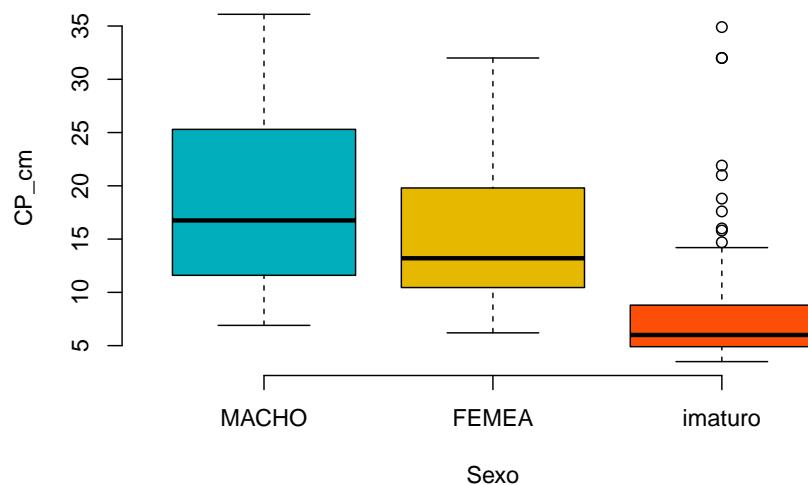
```

ggline(univ, x = "sexo", y = "CP_cm",
       add = c("mean_se", "jitter"),
       order = c("MACHO", "FEMEA", "imaturo"),
       ylab = "Weight", xlab = "Treatment")

```



```
boxplot(CP_cm ~ sexo, data = univ,
        xlab = "Sexo", ylab = "CP_cm",
        frame = FALSE, col = c("#00AFBB", "#E7B800", "#FC4E07"))
```



```
library(gplots)

## 
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
## 
##     lowess

plotmeans(CP_cm ~ sexo, data = univ, frame = FALSE,
          xlab = "Sexo", ylab = "CP_cm",
          main="Mean Plot with 95% CI")

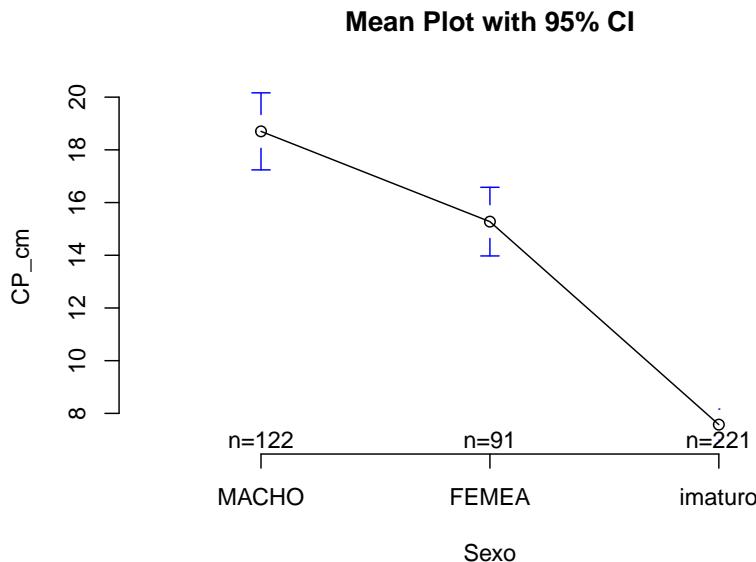
## Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "frame" não é um
## parâmetro gráfico

## Warning in axis(1, at = 1:length(means), labels = legends, ...): "frame" não é
## um parâmetro gráfico

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "frame" não é um
## parâmetro gráfico
```



```
# ANOVA
anova <- aov(CP_cm ~ sexo, data = univ)
summary(anova)
#Interpretação: Um valor de p MENOR que o nível de significância de 0.05 significa que
#EXISTE diferença significativa entre as três grupos de médias.

# Comparações múltiplas
TukeyHSD(anova)
library(multcomp)

## Warning: package 'multcomp' was built under R version 4.3.3
## Carregando pacotes exigidos: mvtnorm
## Carregando pacotes exigidos: survival
## Carregando pacotes exigidos: TH.data
## Warning: package 'TH.data' was built under R version 4.3.3
## Carregando pacotes exigidos: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##       select
```

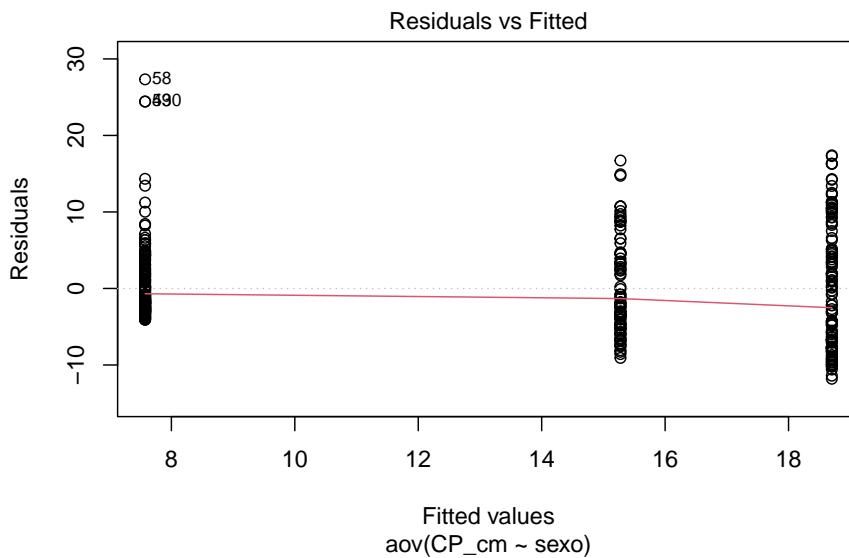
```

## 
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
## 
##     geyser

library(multcomp)
summary(glht(anova, linfct = mcp(sexo = "Tukey")))
# T-test entre pares
pairwise.t.test(univ$CP_cm, univ$sexo,
                p.adjust.method = "BH")
# Pressupostos
## Homogeneidade de variâncias
plot(anova, 1) #valores numerados são outliers

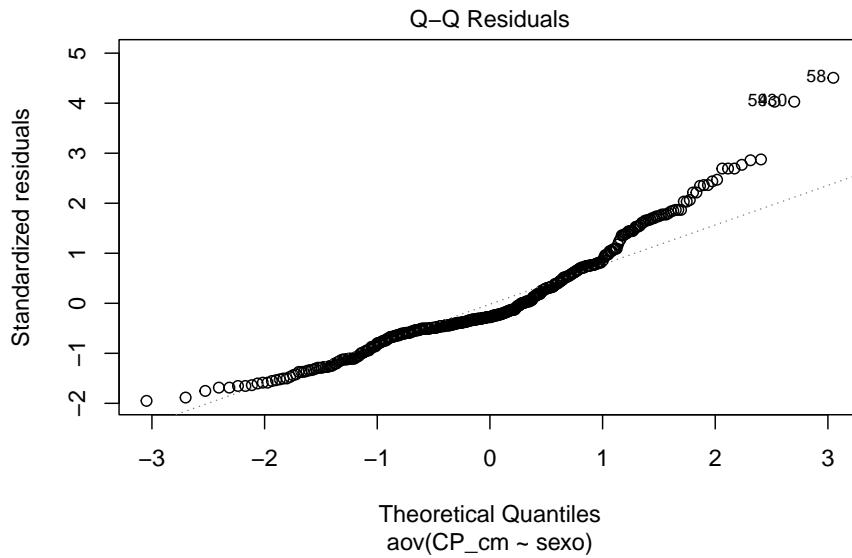
```



```

library(car)
leveneTest(CP_cm ~ sexo, data = univ)
# ANOVA sem o pressuposto de equalidade de variâncias
oneway.test(CP_cm ~ sexo, data = univ)
# Testes pareados sem o pressuposto de equalidade de variâncias
pairwise.t.test(univ$CP_cm, univ$sexo,
                p.adjust.method = "BH", pool.sd = FALSE)
# Normalidade pelos resíduos (Q-Q plot)
plot(anova, 2)

```



```
# Extraindo os resíduos e rodando o Shapiro-Wilk neles
anova_residuals <- residuals(object = anova)
shapiro.test(x = anova_residuals)
# ANOVA não-paramétrica (Kruskal-Wallis)
kruskal.test(CP_cm ~ sexo, data = univ)
```

```
## [1] "MACHO"    "FEMEA"     "imaturo"
## # A tibble: 3 x 4
##   sexo      count   mean    sd
##   <ord>     <int> <dbl> <dbl>
## 1 MACHO      122 18.7  8.16
## 2 FEMEA       91 15.3  6.25
## 3 imaturo    221  7.57  4.44
##             Df Sum Sq Mean Sq F value Pr(>F)
## sexo          2 10743   5371   145.5 <2e-16 ***
## Residuals    431 15911      37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = CP_cm ~ sexo, data = univ)
```

```

## 
## $sexo
##          diff      lwr      upr     p adj
## FEMEA-MACHO -3.424716 -5.403982 -1.445451 0.0001655
## imaturo-MACHO -11.127884 -12.739590 -9.516177 0.0000000
## imaturo-FEMEA -7.703167 -9.482984 -5.923350 0.0000000
##
## 
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
## 
## Fit: aov(formula = CP_cm ~ sexo, data = univ)
##
## Linear Hypotheses:
##             Estimate Std. Error t value Pr(>|t|)    
## FEMEA - MACHO == 0 -3.4247    0.8416 -4.069 0.000161 ***
## imaturo - MACHO == 0 -11.1279   0.6853 -16.238 < 1e-04 ***
## imaturo - FEMEA == 0 -7.7032   0.7568 -10.179 < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
##
## 
## Pairwise comparisons using t tests with pooled SD
##
## data:  univ$CP_cm and univ$sexo
##
##          MACHO    FEMEA
## FEMEA  5.6e-05 -
## imaturo < 2e-16 < 2e-16
##
## P value adjustment method: BH
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)    
## group     2  45.138 < 2.2e-16 ***
##           431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## One-way analysis of means (not assuming equal variances)
##
## data:  CP_cm and sexo
## F = 134.26, num df = 2.00, denom df = 180.92, p-value < 2.2e-16
##

```

```

## 
##  Pairwise comparisons using t tests with non-pooled SD
## 
##  data:  univ$CP_cm and univ$sexo
## 
##          MACHO    FEMEA
## FEMEA   0.00063 -
## imaturo < 2e-16 < 2e-16
## 
## P value adjustment method: BH
## 
##  Shapiro-Wilk normality test
## 
##  data:  anova_residuals
## W = 0.93021, p-value = 2.35e-13
## 
## 
##  Kruskal-Wallis rank sum test
## 
##  data:  CP_cm by sexo
## Kruskal-Wallis chi-squared = 217.39, df = 2, p-value < 2.2e-16

```

9.7.4 Teste entre três médias e dois fatores (Two-Way ANOVA)

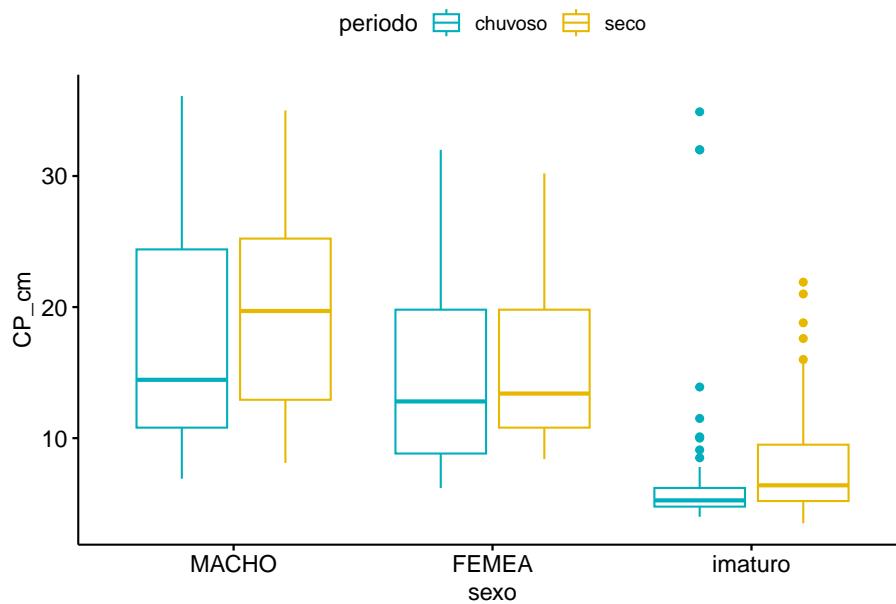
Two-way ANOVA

```

univ$sexo <- factor(univ$sexo,
                      labels = c("MACHO", "FEMEA", "imaturo"))
univ$periodo <- factor(univ$periodo,
                        labels = c("chuvaso", "seco"))
str(univ)
table(univ$sexo,univ$periodo)

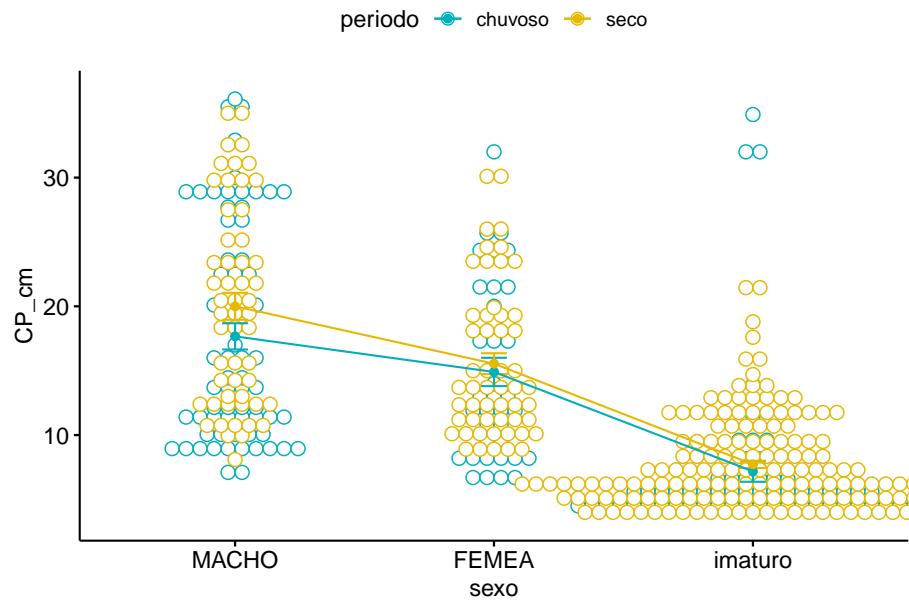
library("ggpubr")
ggboxplot(univ, x = "sexo", y = "CP_cm", color = "periodo",
          palette = c("#00AFBB", "#E7B800"))

```

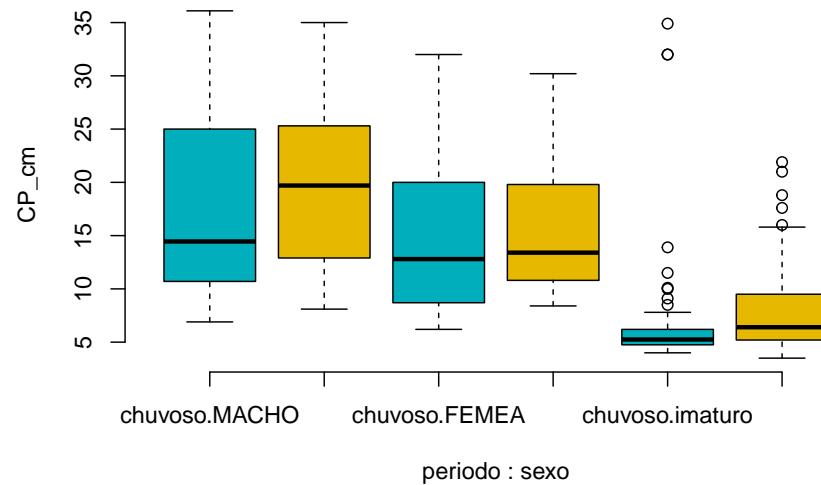


```
ggline(univ, x = "sexo", y = "CP_cm", color = "periodo",
       add = c("mean_se", "dotplot"),
       palette = c("#00AFBB", "#E7B800"))
```

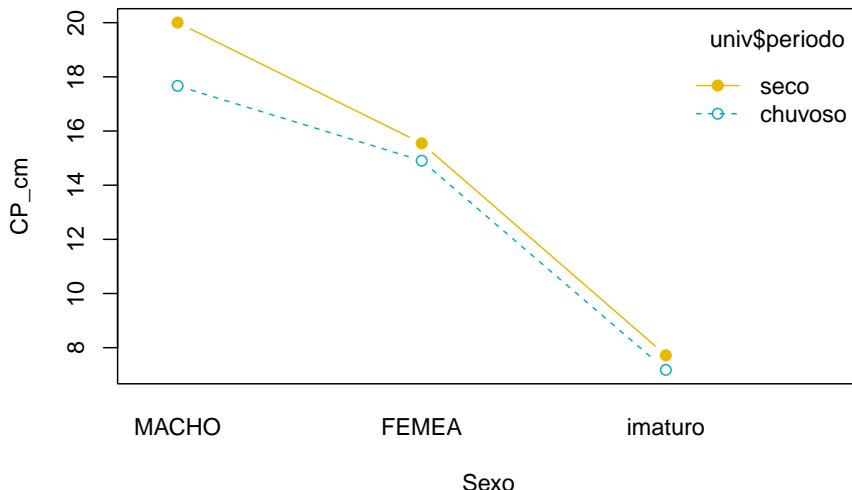
```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



```
boxplot(CP_cm ~ periodo * sexo, data=univ, frame = FALSE,
        col = c("#00AFBB", "#E7B800"), ylab="CP_cm")
```



```
# Gráfico das interações
interaction.plot(x.factor = univ$sexo, trace.factor = univ$periodo,
                 response = univ$CP_cm, fun = mean,
                 type = "b", legend = TRUE,
                 xlab = "Sexo", ylab="CP_cm",
                 pch=c(1,19), col = c("#00AFBB", "#E7B800"))
```



```
# ANOVA Two-way
anova2 <- aov(CP_cm ~ periodo + sexo, data = univ)
summary(anova2)

#The above-fitted model is not referred to as an additive model.
#It is presumptively assumed that the two-factor variables are unrelated.
#Replace the plus symbol (+) with an asterisk (*) if you think
#these two variables will interact to create a synergistic effect.

anova3 <- aov(CP_cm ~ periodo * sexo, data = univ)
summary(anova3)

require("dplyr")
group_by(univ, periodo, sexo) %>%
  summarise(
    count = n(),
```

```

mean = mean(CP_cm, na.rm = TRUE),
sd = sd(CP_cm, na.rm = TRUE)
)

## `summarise()` has grouped output by 'periodo'. You can override using the
## `.groups` argument.

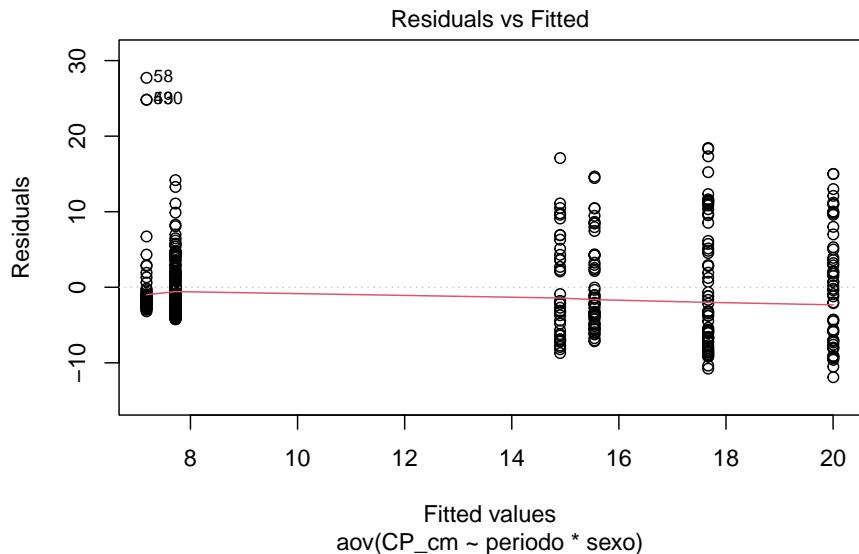
# Comparações múltiplas pares: Tukey
TukeyHSD(anova3, which = "sexo")

# Comparações múltiplas pares
library(multcomp)
summary(glht(anova2, linfct = mcp(sexo = "Tukey")))

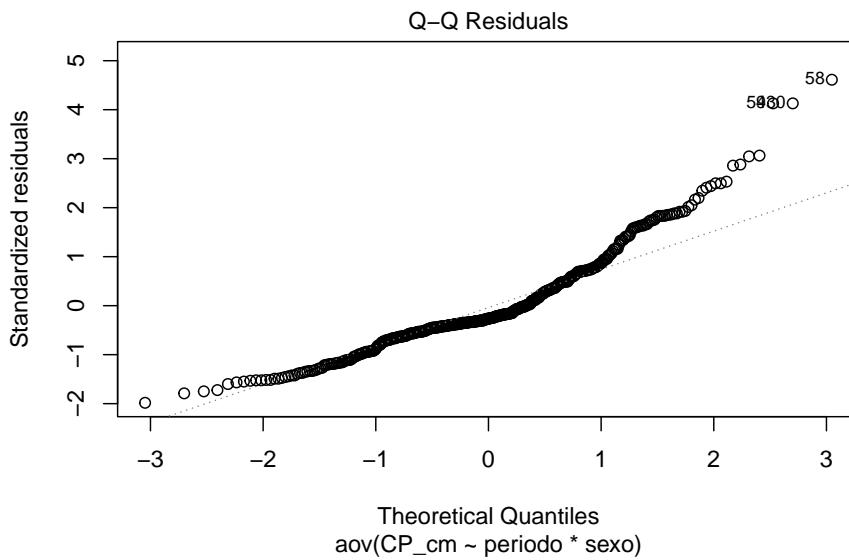
# T-test entre pares
pairwise.t.test(univ$CP_cm, univ$sexo,
                p.adjust.method = "BH")

# Pressupostos
## Homogeneidade de variâncias
plot(anova3, 1)

```



```
library(car)
leveneTest(CP_cm ~ periodo*sexo, data = univ)
# Normalidade pelos resíduos (Q-Q plot)
plot(anova3, 2)
```



```
# Extraindo os resíduos e rodando o Shapiro-Wilk neles
anova3_residuals <- residuals(object = anova3)
shapiro.test(x = anova3_residuals )

# ANOVA para desenhos não balanceados
library(car)
nb_anova <- aov(CP_cm ~ periodo * sexo, data = univ)
Anova(nb_anova, type = "III")
```

```
## 'data.frame': 434 obs. of 23 variables:
## $ n.=.434 : chr "TU001" "TU002" "TU003" "TU004" ...
## $ CT_cm : num 32.4 33.4 27.3 13.2 14.3 22.7 23.2 13.5 24.6 19.4 ...
## $ PT_g : num 468.8 520 301.5 28.2 38.9 ...
## $ CP_cm : num 27.2 28.8 23.8 11 11.9 20.5 19.2 11.5 20.5 16 ...
## $ Ctubo_cm: num 39.8 14.3 13 16.5 15.5 24.2 25 17.5 25.7 18.3 ...
## $ PC_g : num 458.9 507.4 283.4 27.7 37.7 ...
## $ %PT : num 2.11 2.42 6 1.77 3.08 ...
## $ Pest_g : num 3.9 5.9 15.5 0.3 0.8 9.9 6.8 4.1 2.5 0.7 ...
```

```

## $ Cest_cm : num 7.7 10 10.2 4.3 4.5 6.4 7.5 6.8 7.6 5.1 ...
## $ gr_est : chr "I" "I" "III" "I" ...
## $ ir_est : num 0.832 1.135 5.141 1.064 2.057 ...
## $ Pint_g : num 4.8 5.3 2.4 0.1 0.3 2.7 3.3 0.4 2 1.4 ...
## $ Cint_cm : num 38.3 13.3 12 16 15 23.7 24.5 17 25.3 17.7 ...
## $ gr_int : chr "II" "II" "II" "II" ...
## $ ir_int : num 1.024 1.019 0.796 0.355 0.771 ...
## $ Pgon_g : num 1.2 1.4 0.2 0.1 0.1 0.4 13.8 0.1 0.1 0.1 ...
## $ Cgon_cm : chr "7" "6.5" "7.7" "5" ...
## $ emg : chr "IMATURO" "MADURO" "EM MATERACAO" "IMATURO" ...
## $ ig : num 0.256 0.2692 0.0663 0.3546 0.2571 ...
## $ mes : chr "ago" "ago" "ago" "ago" ...
## $ periodo : Factor w/ 2 levels "chuvisco","seco": 1 1 1 1 1 1 1 1 1 ...
## $ estação : chr "inverno" "inverno" "inverno" "inverno" ...
## $ sexo : Ord.factor w/ 3 levels "MACHO"><"FEMEA"><...: 1 1 1 1 1 2 3 1 1 ...
##
##          chuvisco seco
## MACHO      68   54
## FEMEA      38   53
## imaturo    60  161
##             Df Sum Sq Mean Sq F value Pr(>F)
## periodo     1   231     231    6.29 0.0125 *
## sexo        2 10634     5317 144.80 <2e-16 ***
## Residuals  430 15789      37
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##             Df Sum Sq Mean Sq F value Pr(>F)
## periodo     1   231     231    6.287 0.0125 *
## sexo        2 10634     5317 144.714 <2e-16 ***
## periodo:sexo 2     64      32    0.874 0.4180
## Residuals  428 15725      37
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## # A tibble: 6 x 5
## # Groups: periodo [2]
## periodo sexo count mean   sd
## <fct>   <ord>  <int> <dbl> <dbl>
## 1 chuvisco MACHO      68 17.7   8.41
## 2 chuvisco FEMEA     38 14.9   6.78
## 3 chuvisco imaturo    60 7.18   6.25
## 4 seco      MACHO      54 20.0   7.70
## 5 seco      FEMEA     53 15.5   5.90
## 6 seco      imaturo    161 7.72  3.56
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##

```

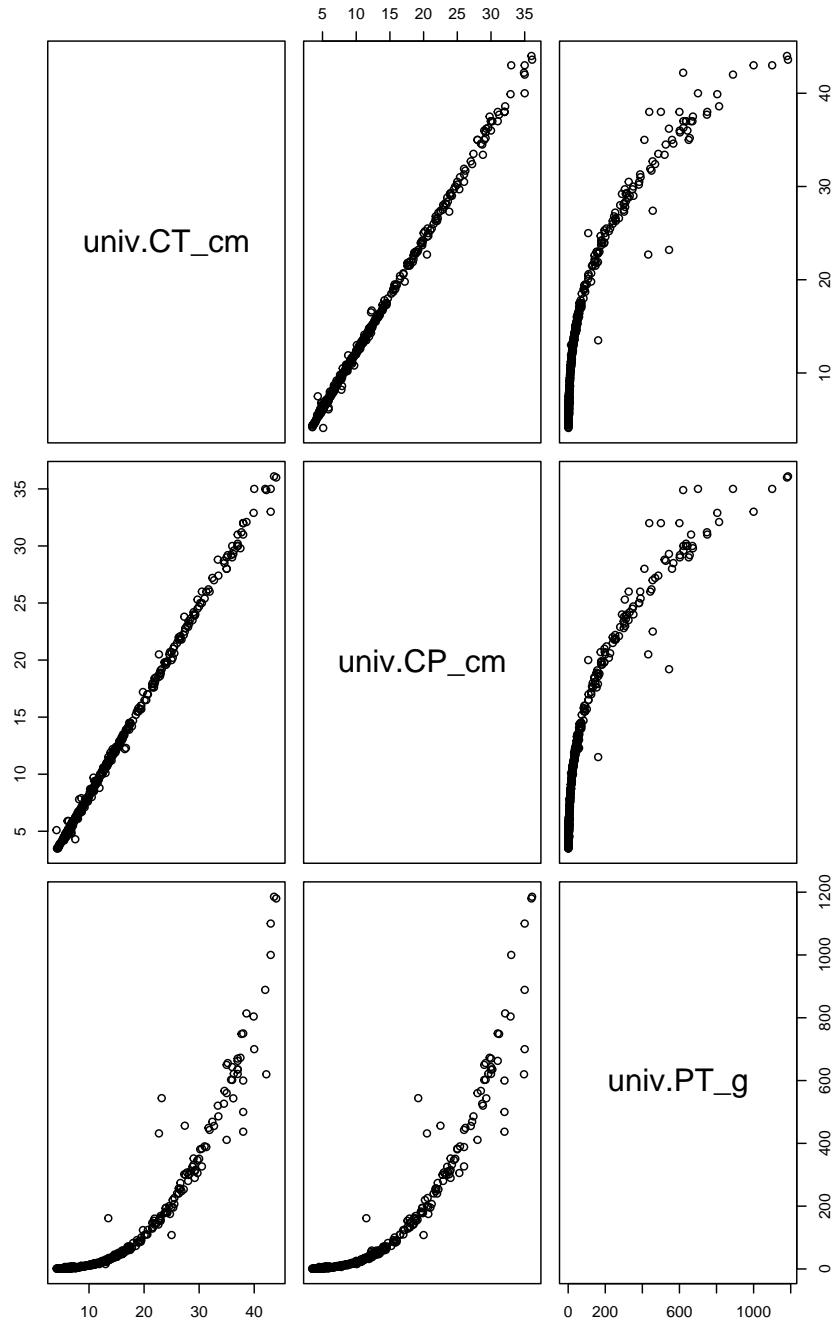
```
## Fit: aov(formula = CP_cm ~ periodo * sexo, data = univ)
##
## $sexo
##          diff      lwr      upr     p adj
## FEMEA-MACHO -3.214875 -5.189460 -1.240291 0.0004334
## imaturo-MACHO -10.698753 -12.306648 -9.090859 0.0000000
## imaturo-FEMEA -7.483878 -9.259486 -5.708270 0.0000000
##
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = CP_cm ~ periodo + sexo, data = univ)
##
## Linear Hypotheses:
##                         Estimate Std. Error t value Pr(>|t|)
## FEMEA - MACHO == 0    -3.5823    0.8438 -4.246 7.4e-05 ***
## imaturo - MACHO == 0 -11.4502    0.7060 -16.219 < 1e-05 ***
## imaturo - FEMEA == 0 -7.8679    0.7601 -10.351 < 1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
##
##
## Pairwise comparisons using t tests with pooled SD
##
## data:  univ$CP_cm and univ$sexo
##
##          MACHO    FEMEA
## FEMEA  5.6e-05 -
## imaturo < 2e-16 < 2e-16
##
## P value adjustment method: BH
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group    5 16.044 1.67e-14 ***
##          428
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Shapiro-Wilk normality test
##
## data:  anova3_residuals
## W = 0.92359, p-value = 4.653e-14
```

```
##  
## Anova Table (Type III tests)  
##  
## Response: CP_cm  
##           Sum Sq Df  F value Pr(>F)  
## (Intercept) 27390.0  1 745.5129 <2e-16 ***  
## periodo      122.3  1   3.3281 0.0688 .  
## sexo         3641.5  2   49.5584 <2e-16 ***  
## periodo:sexo  64.2  2   0.8741 0.4180  
## Residuals    15724.6 428  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.8 Correlação e regressão simples

Correlação e regressão

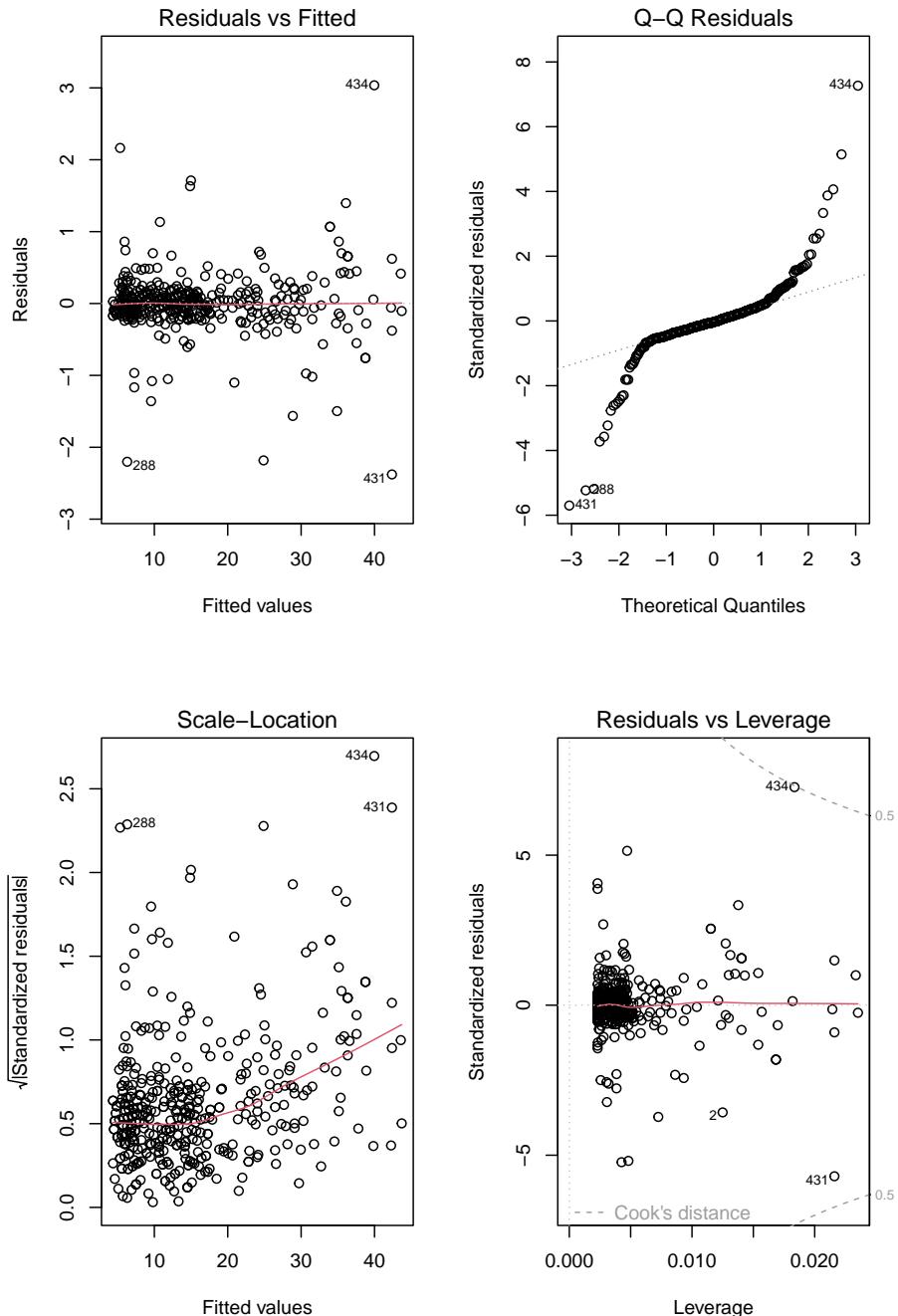
```
df<-data.frame(univ$CT_cm, univ$CP_cm, univ$PT_g)  
plot(df[,1:3])
```



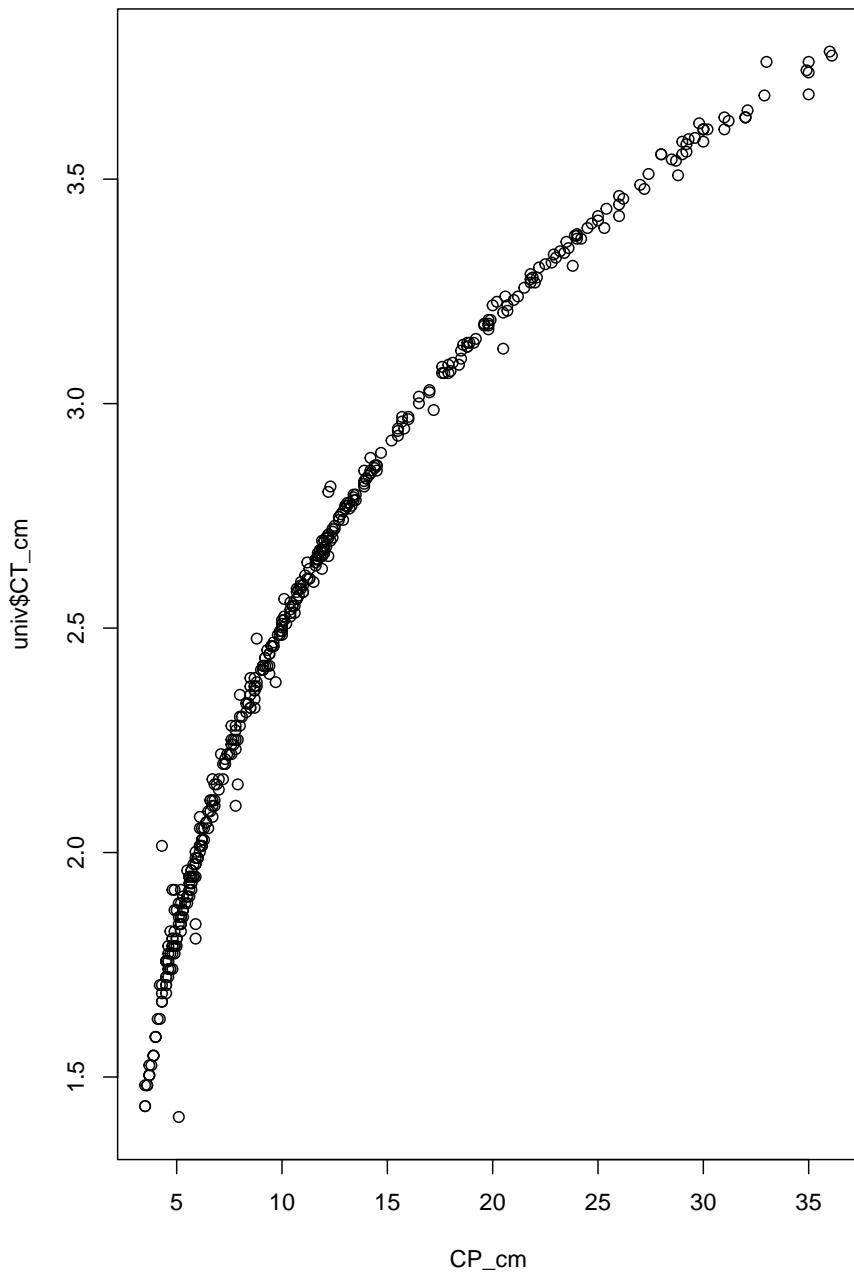
```
cor(df,method="pearson")
cor(df[,1:3], method="spearman")
cor.test(univ$CT_cm, univ$CP_cm, method="pearson")
cor.test(univ$CT_cm, univ$CP_cm, method="spearman")
```

```
## Warning in cor.test.default(univ$CT_cm, univ$CP_cm, method = "spearman"):
## Impossível calcular o valor exato de p com empates

m1 <- lm(univ$CT_cm ~ univ$CP_cm, data=univ)
summary(m1)
par(mfrow=c(2,2))
plot(m1)
```



```
par(mfrow=c(1,1))
univ$CT_cm <- log(univ$CT_cm)
plot(univ$CT_cm ~ CP_cm, univ)
```



```

anova(m1)
library(car)
Anova(m1)
Anova(m1, white.adjust=TRUE)

## Coefficient covariances computed by hccm()

univ <- univ[univ$CT_cm!=max(univ$CT_cm),]

##          univ.CT_cm univ.CP_cm univ.PT_g
## univ.CT_cm  1.0000000  0.9990120  0.9057485
## univ.CP_cm   0.9990120  1.0000000  0.9039503
## univ.PT_g    0.9057485  0.9039503  1.0000000
##          univ.CT_cm univ.CP_cm univ.PT_g
## univ.CT_cm  1.0000000  0.9976789  0.9924823
## univ.CP_cm   0.9976789  1.0000000  0.9934510
## univ.PT_g    0.9924823  0.9934510  1.0000000
##
## Pearson's product-moment correlation
##
## data: univ$CT_cm and univ$CP_cm
## t = 467.22, df = 432, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9988068 0.9991819
## sample estimates:
##      cor
## 0.999012
##
##
## Spearman's rank correlation rho
##
## data: univ$CT_cm and univ$CP_cm
## S = 31624, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.9976789
##
##
## Call:
## lm(formula = univ$CT_cm ~ univ$CP_cm, data = univ)
##
## Residuals:
```

```
##      Min     1Q Median     3Q    Max
## -2.3777 -0.1281 -0.0147  0.1260  3.0354
##
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.147883  0.037700  3.923 0.000102 ***
## univ$CP_cm  1.206566  0.002582 467.221 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4216 on 432 degrees of freedom
## Multiple R-squared:  0.998, Adjusted R-squared:  0.998
## F-statistic: 2.183e+05 on 1 and 432 DF,  p-value: < 2.2e-16
##
## Analysis of Variance Table
##
## Response: univ$CT_cm
##             Df Sum Sq Mean Sq F value    Pr(>F)
## univ$CP_cm   1 38802  38802 218295 < 2.2e-16 ***
## Residuals 432     77       0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Anova Table (Type II tests)
##
## Response: univ$CT_cm
##             Sum Sq Df F value    Pr(>F)
## univ$CP_cm  38802   1 218295 < 2.2e-16 ***
## Residuals     77 432
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Analysis of Deviance Table (Type II tests)
##
## Response: univ$CT_cm
##             Df F    Pr(>F)
## univ$CP_cm   1 75315 < 2.2e-16 ***
## Residuals 432
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Apêndices

Sites consultados

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executa-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

Referências

Chapter 10

R Modulo 3.3 - Gráficos univariados

RESUMO

Apresentação

10.1 Sobre os dados

Considere os dados merísticos (ou medições morfológicas) da espécie de peixe *Cichla ocellaris* (tucunaré amarelo) do reservatório da barragem de Gramame, PB (MEDEIROS; ROSA, 1994) (Figura 10.1). Existem 421 medições do comprimento total (CT), comprimento padrão (CP) e peso total (PT), além do sexo (MACHO, FÊMEA ou imaturo), e outros descritores da estrutura populacional da espécie, um conjunto de dados formidável.

10.2 Organização básica

```
dev.off() #apaga os graficos, se houver algum  
rm(list=ls(all=TRUE)) #limpa a memória  
cat("\014") #limpa o console
```

Instalando os pacotes necessários para esse módulo

```
install.packages("mdatools")
```

n = 421	mes	periodo	estação	sexo	CT_cm	CP_cm	PT_g
TU 0001	ago	chuvisco	inverno	MACHO	32,4	27,2	468,8
TU 0002	ago	chuvisco	inverno	MACHO	33,4	28,8	520,0
TU 0003	ago	chuvisco	inverno	MACHO	27,3	23,8	301,5
TU 0004	ago	chuvisco	inverno	MACHO	13,2	11,0	28,2
TU 0005	ago	chuvisco	inverno	MACHO	14,3	11,9	38,9
TU 0006	set	chuvisco	inverno	MACHO	22,7	20,5	431,7
TU 0007	set	chuvisco	inverno	FEMEA	23,2	19,2	544,0
TU 0008	set	chuvisco	inverno	imaturo	13,5	11,5	161,6
TU 0009	set	chuvisco	inverno	MACHO	24,6	20,5	200,5
TU 0010	set	chuvisco	inverno	MACHO	19,4	16,0	86,7
TU 0011	set	chuvisco	inverno	FEMEA	25,5	21,2	206,7
TU 0012	set	chuvisco	inverno	MACHO	23,0	18,9	153,0
TU 0013	set	chuvisco	inverno	MACHO	17,4	14,4	68,7
TU 0014	set	chuvisco	inverno	FEMEA	16,5	12,2	53,6
TU 0015	set	chuvisco	inverno	MACHO	21,9	18,4	141,1
TU 0016	set	chuvisco	inverno	MACHO	28,1	23,4	299,3
TU 0017	set	chuvisco	inverno	MACHO	27,2	22,2	253,6
TU 0018	set	chuvisco	inverno	MACHO	25,5	20,6	226,0
TU 0019	set	chuvisco	inverno	FEMEA	38,0	32,0	437,1
TU 0020	set	chuvisco	inverno	MACHO	27,5	22,8	299,8
TU 0021	set	chuvisco	inverno	imaturo	7,8	6,2	4,5
TU 0022	set	chuvisco	inverno	imaturo	6,6	5,1	2,1
TU 0023	set	chuvisco	inverno	imaturo	7,8	6,1	4,0
TU 0024	set	chuvisco	inverno	imaturo	8,3	6,6	5,5
TU 0025	set	chuvisco	inverno	imaturo	8,6	6,8	6,7
TU 0026	set	chuvisco	inverno	imaturo	6,3	5,9	2,5
TU 0027	set	chuvisco	inverno	imaturo	8,0	6,1	4,9
TU 0028	set	chuvisco	inverno	imaturo	10,9	8,5	13,5
TU 0029	set	chuvisco	inverno	imaturo	7,1	5,5	3,3
TU 0030	set	chuvisco	inverno	imaturo	9,8	7,8	10,0
TU 0031	set	chuvisco	inverno	imaturo	6,0	4,6	2,1
TU 0032	set	chuvisco	inverno	imaturo	6,1	5,9	2,0
TU 0033	set	chuvisco	inverno	imaturo	6,2	4,9	1,9
TU 0034	set	chuvisco	inverno	imaturo	6,6	5,2	2,3
TU 0035	set	chuvisco	inverno	imaturo	6,5	5,0	2,5
TU 0036	set	chuvisco	inverno	imaturo	6,1	4,8	2,2
TU 0037	set	chuvisco	inverno	imaturo	6,1	4,8	2,5
TU 0038

Figure 10.1: Dados merísticos da espécie de peixe **Cichla ocellaris** (tucunaré amarelo) do reservatório da barragem de Gramame, PB.

```
library(openxlsx)
```

Os códigos acima, são usados para instalar e carregar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades do R (R CRAN (TEAM, 2017) e RStudio (TEAM, 2022)). No exemplo acima, o pacote `openxlsx` permite ler e escrever arquivos Excel no R. Para instalar um pacote no R, você precisa usar a função `install.packages()`.

Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`. Por exemplo, para carregar o pacote `openxlsx`, você precisa executar a função `library(openxlsx)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

10.3 Importando a planilha



ATENÇÃO

Os links para baixar as planilhas necessárias para repetir esse tutorial podem ser encontrados na seção Arquivos disponíveis do Capítulo Bases de dados.

Ou, baixe aqui o arquivo `tucuna.xlsx`

Vamos importar a planilha de dados univariados `tucuna.xlsx`. Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações. Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

```
library(openxlsx)
univ_all <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/tucuna.xlsx",
```

```

      rowNames = T, colNames = T,
      sheet = "tucuna")
head(univ_all,10)
head(univ_all[, 1:5], 10)

##      CT_cm  PT_g CP_cm Ctubo_cm  PC_g      %PT Pest_g Cest_cm gr_est    ir_est
## TU001  32.4 468.8  27.2     39.8 458.9 2.111775   3.9    7.7      I 0.8319113
## TU002  33.4 520.0  28.8     14.3 507.4 2.423077   5.9   10.0      I 1.1346154
## TU003  27.3 301.5  23.8     13.0 283.4 6.003317  15.5   10.2     III 5.1409619
## TU004  13.2 28.2   11.0     16.5 27.7 1.773050   0.3    4.3      I 1.0638298
## TU005  14.3 38.9   11.9     15.5 37.7 3.084833   0.8    4.5     III 2.0565553
## TU006  22.7 431.7  20.5     24.2 418.7 3.011350   9.9    6.4     III 2.2932592
## TU007  23.2 544.0  19.2     25.0 520.1 4.393382   6.8    7.5      II 1.2500000
## TU008  13.5 161.6  11.5     17.5 157.0 2.846535   4.1    6.8      II 2.5371287
## TU009  24.6 200.5  20.5     25.7 195.9 2.294264   2.5    7.6      II 1.2468828
## TU010  19.4 86.7   16.0     18.3 84.5 2.537486   0.7    5.1      I 0.8073818
##      Pint_g Cint_cm gr_int    ir_int Pgon_g Cgon_cm          emg          ig
## TU001   4.8   38.3   II 1.0238908   1.2    7  IMATURO 0.25597270
## TU002   5.3   13.3   II 1.0192308   1.4   6.5  MADURO 0.26923077
## TU003   2.4   12.0   II 0.7960199   0.2   7.7 EM MATURACAO 0.06633499
## TU004   0.1   16.0   II 0.3546099   0.1    5  IMATURO 0.35460993
## TU005   0.3   15.0   II 0.7712082   0.1   3.2  IMATURO 0.25706941
## TU006   2.7   23.7   II 0.6254343   0.4   7.8 EM MATURACAO 0.09265694
## TU007   3.3   24.5   II 0.6066176  13.8   7.9  MADURO 2.53676471
## TU008   0.4   17.0    I 0.2475248   0.1   2.5  IMATURO 0.06188119
## TU009   2.0   25.3   II 0.9975062   0.1   6.5 EM MATURACAO 0.04987531
## TU010   1.4   17.7   II 1.6147636   0.1    6  IMATURO 0.11534025
##      mes periodo estação    sexo
## TU001 ago chuvoso inverno MACHO
## TU002 ago chuvoso inverno MACHO
## TU003 ago chuvoso inverno MACHO
## TU004 ago chuvoso inverno MACHO
## TU005 ago chuvoso inverno MACHO
## TU006 set chuvoso inverno MACHO
## TU007 set chuvoso inverno FEMEA
## TU008 set chuvoso inverno imaturo
## TU009 set chuvoso inverno MACHO
## TU010 set chuvoso inverno MACHO
##      CT_cm  PT_g CP_cm Ctubo_cm  PC_g
## TU001  32.4 468.8  27.2     39.8 458.9
## TU002  33.4 520.0  28.8     14.3 507.4
## TU003  27.3 301.5  23.8     13.0 283.4
## TU004  13.2 28.2   11.0     16.5 27.7
## TU005  14.3 38.9   11.9     15.5 37.7
## TU006  22.7 431.7  20.5     24.2 418.7

```

```
## TU007 23.2 544.0 19.2      25.0 520.1
## TU008 13.5 161.6 11.5      17.5 157.0
## TU009 24.6 200.5 20.5      25.7 195.9
## TU010 19.4 86.7 16.0       18.3 84.5
```

Exibindo os dados importados (esses comando são “case-sensitive” `ignore.case(object)`).

```
#View(univ)
print(univ_all[1:5,1:5])
univ_all
str(univ_all)
mode(univ_all)
class(univ_all)
```

10.4 Reset point

```
univ <- univ_all
univ <- univ_all[, 1:3]
#univ_all <- univ_all
#var <- ""
```

10.5 Gráficos de dispersão

(<https://mda.tools/docs/datasets--simple-plots.html>)

```
library(mdatools)
#removendo linhas ou coluns por nome
attr(univ, "name") = "Tucunaré"
attr(univ, "xaxis.name") = "Eixo X"
attr(univ, "yaxis.name") = "Eixo Y"
par(mfrow = c(1,2))
# show plot for the whole dataset (columns 1 and 2 will be taken)
mdaplot(univ, type = "p")
# subset the dataset and keep only columns 6 and 7 and then make a plot
mdaplot(mda.subset(univ, select = c(1,2)), type = "p")

par(mfrow = c(2,2))
# show Height vs Weight and color points by the Beer consumption
mdaplot(univ, type = "p", cgroup = univ_all[, var])
# do the same but do not show colorbar
mdaplot(univ, type = "p", cgroup = univ_all[, var],
        show.colorbar = FALSE)
# do the same but use grayscale color map
```

```

mdaplot(univ, type = "p", cgroup = univ_all[, var], colormap = "gray")
# do the same but using colormap with gradients between red, yellow and green colors
mdaplot(univ, type = "p", cgroup = univ_all[, var],
        colormap = c("red", "yellow", "green"))
# make a factor using values of variable Sex and define labels for the factor levels
g <- factor(univ_all[, "sexo"],
             levels = c("MACHO", "FEMEA", "imaturo"))
g
par(mfrow = c(1, 2))
mdaplot(univ, type = "p", cgroup = g)
mdaplot(univ, type = "p", cgroup = g, colormap = "gray")

par(mfrow = c(1, 2))
# default way - color grouping is used for borders and "bg" for background
mdaplot(univ, type = "p", cgroup = univ_all[, var], pch = 21, bg = "white")
# inverse - color grouping is used for background and "bg" for border
mdaplot(univ, type = "p", cgroup = univ_all[, var], pch = 21, bg = "white", pch.colinv = TRUE)

par(mfrow = c(2, 2))
# by default row names will be used as labels
mdaplot(univ, type = "p", show.labels = TRUE)
# here we tell to use indices as labels instead
mdaplot(univ, type = "p", show.labels = TRUE, labels = "indices")
# here we use names again but change color and size of the labels
mdaplot(univ, type = "p", show.labels = TRUE, labels = "names", lab.col = "red", lab.cex = 2)
# finally we provide a vector with manual values to be used as the labels
mdaplot(univ, type = "p", show.labels = TRUE, labels = paste0("T", seq_len(nrow(univ)))))

par(mfrow = c(2, 2))
# manual values and tick labels for the x-axis
mdaplot(univ, xticks = c(10,30,40), xticklabels = c("Small", "Medium", "Large"))
# same but with rotation of the tick labels
mdaplot(univ, xticks = c(10,30,40), xticklabels = c("Small", "Medium", "Large"), xlas = 2)
# manual values and tick labels for the y-axis
mdaplot(univ, yticks = c(200,600,900), yticklabels = c("Light", "Medium", "Heavy"))
# same but with rotation of the tick labels
mdaplot(univ, yticks = c(200,600,900), yticklabels = c("Light", "Medium", "Heavy"), ylas = 2)

par(mfrow = c(1, 2))
# change margin for bottom part
par(mar = c(6, 4, 4, 2) + 0.1)
mdaplot(univ, xticks = c(10,30,40),
        xticklabels = c("Small", "Medium", "Large"),
        xlas = 2, xlab = "")

```

```

mtext("Comprimento", side = 1, line = 5)
# change margin for left part
par(mar = c(5, 6, 4, 1) + 0.1)
mdaplot(univ, yticks = c(200,600,900), yticklabels = c("Light", "Medium", "Heavy"),
        ylas = 2, ylab = ""))
mtext("Peso", side = 2, line = 5)

par(mfrow = c(1,2))
mdaplot(univ, show.grid = FALSE, show.lines = c(20,200))
mdaplot(univ, show.lines = c(40, NA))
# define a factor using values of variable Sex and simple labels
g
par(mfrow = c(1, 2))
# make a scatter plot grouping points by the factor and then show convex hull for each group
p <- mdaplot(univ, cgroup = g)
plotConvexHull(p)
# make a scatter plot grouping points by the factor and then show 90% confidence intervals
p <- mdaplot(univ, cgroup = g)
plotConfidenceEllipse(p, conf.level = 0.90)

# média agrupada por fator
media <- tapply(univ$CT_cm, g, mean)
media
anova <- aov(CT_cm ~ g, data = univ_all)
summary(anova)
plot(anova,1)

```

10.6 Gráficos de linha

```

library(mdatoools)

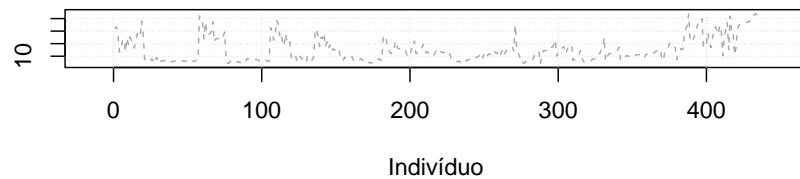
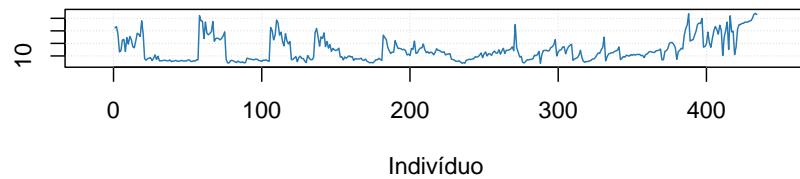
## Warning: package 'mdatoools' was built under R version 4.3.3
##
## Attaching package: 'mdatoools'

## The following object is masked from 'package:car':
##
##     ellipse

var <- univ$CT_cm
attr(univ$CT_cm, "yaxis.name") = "CT_cm"
attr(univ$CT_cm, "xaxis.name") = "Indivíduo"

```

```
par(mfrow = c(2, 1))
mdaplot(univ$CT_cm, type = "l")
mdaplot(univ$CT_cm, type = "l", col = "darkgray", lty = 2)
```



```
#mdaplot(univ$CT_cm, type = "l", cgroup = g)
```

10.7 Gráficos de barra

```
# make a simple two rows matrix with values
d = rbind(
  c(20, 50, 60, 90),
  c(14, 45, 59, 88)
)

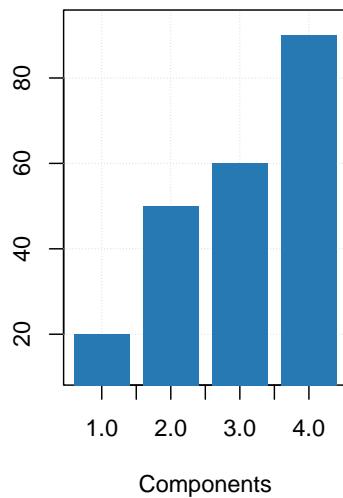
# add some names and attributes
colnames(d) = paste0("PC", 1:4)
rownames(d) = c("Cal", "CV")
attr(d, "xaxis.name") = "Components"
attr(d, "name") = "Explained variance"

par(mfrow = c(1, 2))
```

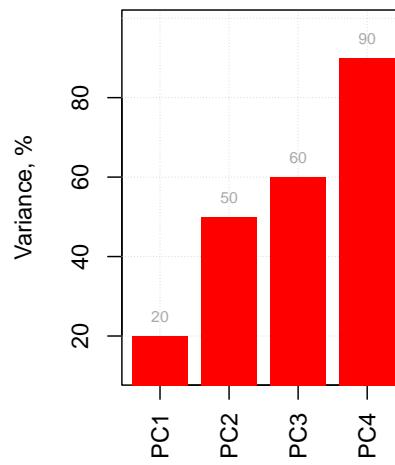
```
# make a default bar plot
mdaplot(d, type = "h")

# make a bar plot with manual xtick labels, color and labels for data values
mdaplot(d, type = "h", xticks = seq_len(ncol(d)), xticklabels = colnames(d), col = "red",
        show.labels = TRUE, labels = "values", xlas = 2, xlab = "", ylab = "Variance, %")
```

Explained variance



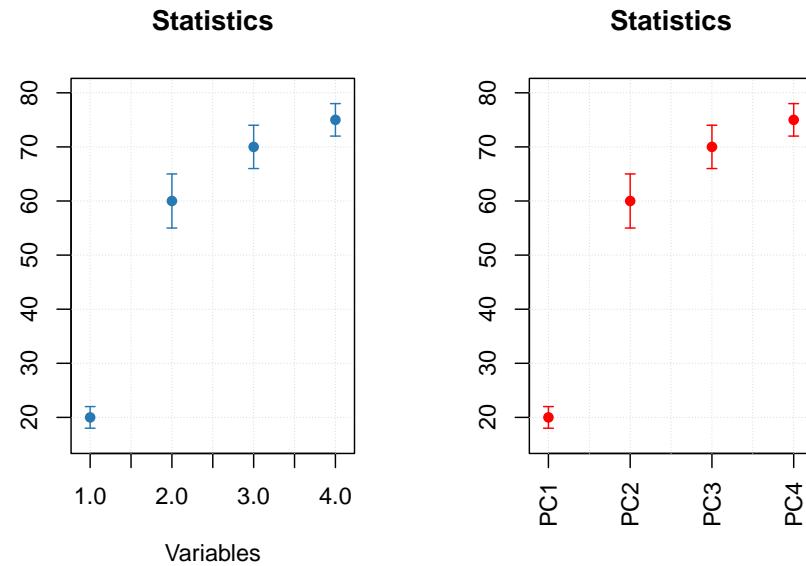
Explained variance



```
d = rbind(
  c(20, 60, 70, 75),
  c(2, 5, 4, 3)
)

# add names and attributes
rownames(d) = c("Mean", "Std")
colnames(d) = paste0("PC", 1:4)
attr(d, 'name') = "Statistics"

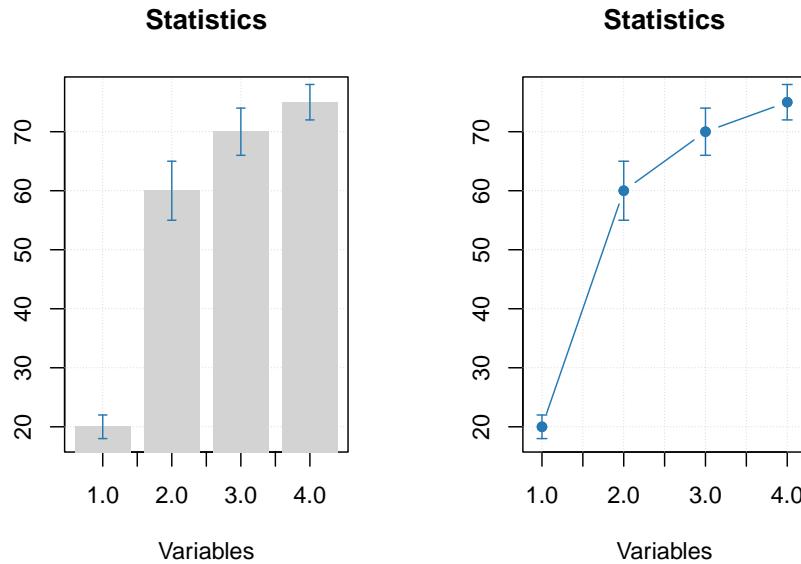
# show the plots
par(mfrow = c(1, 2))
mdaplot(d, type = "e")
mdaplot(d, type = "e", xticks = seq_len(ncol(d)),
        xticklabels = colnames(d), col = "red", xlas = 2, xlab = "")
```



```
par(mfrow = c(1, 2))

mdaplot(mda.subset(d, 1), type = "h", col = "lightgray")
mdaplot(d, type = "e", show.axes = FALSE, pch = NA)

mdaplot(mda.subset(d, 1), type = "b")
mdaplot(d, type = "e", show.axes = FALSE)
```

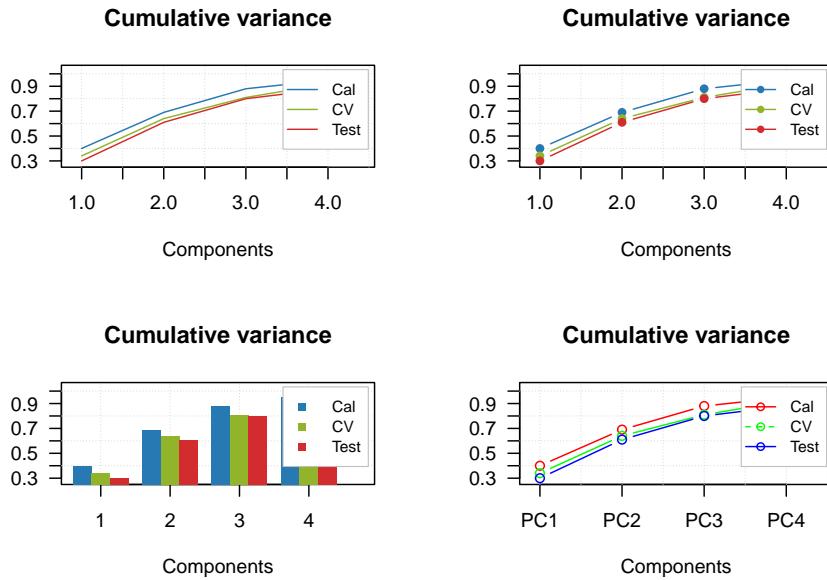


10.8 Gráficos de grupos

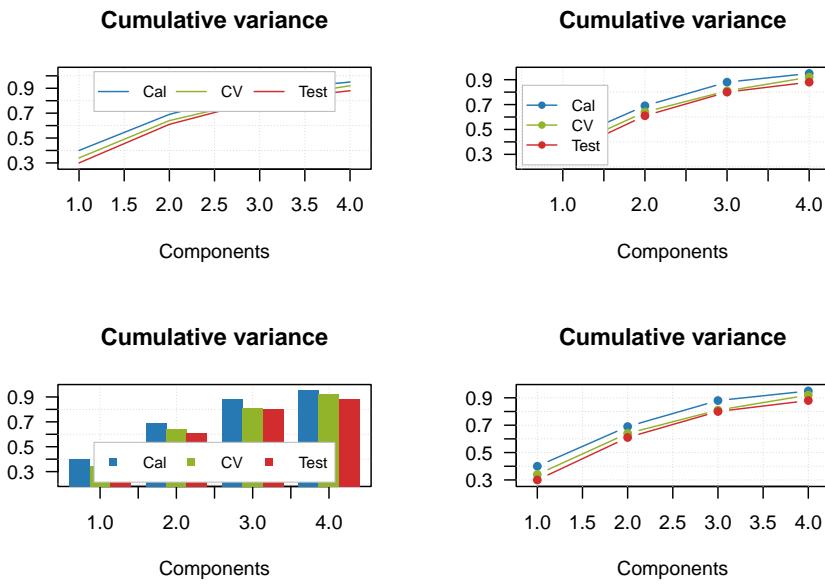
```
# let's create a simple dataset with 3 rows
p = rbind(
  c(0.40, 0.69, 0.88, 0.95),
  c(0.34, 0.64, 0.81, 0.92),
  c(0.30, 0.61, 0.80, 0.88)
)

# add some names and attributes
rownames(p) = c("Cal", "CV", "Test")
colnames(p) = paste0("PC", 1:4)
attr(p, "name") = "Cumulative variance"
attr(p, "xaxis.name") = "Components"

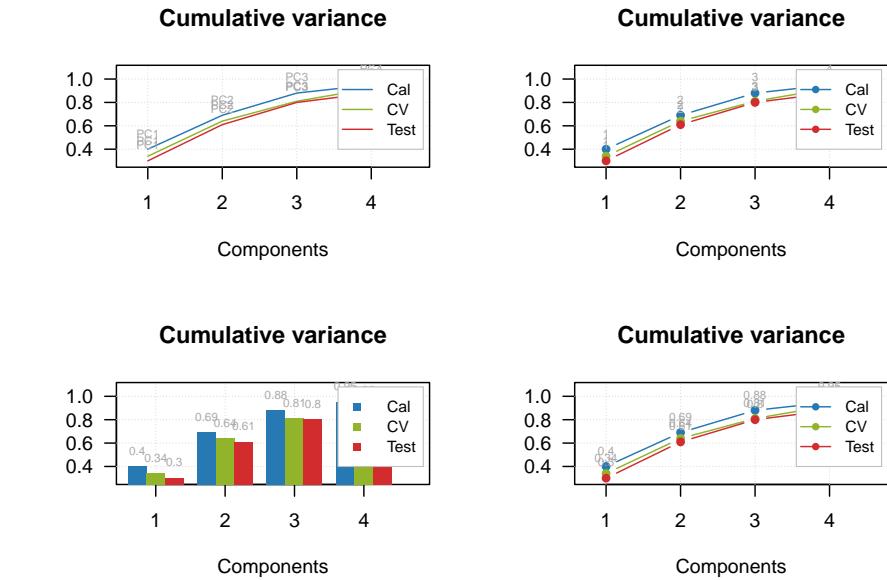
# and make group plots of different types
par(mfrow = c(2, 2))
mdaplotg(p, type = "l")
mdaplotg(p, type = "b")
mdaplotg(p, type = "h", xticks = 1:4)
mdaplotg(p, type = "b", lty = c(1, 2, 1), col = c("red", "green", "blue"), pch = 1,
          xticks = 1:4, xticklabels = colnames(p))
```



```
par(mfrow = c(2, 2))
mdaplotg(p, type = "l", legend.position = "top")
mdaplotg(p, type = "b", legend.position = "bottomleft")
mdaplotg(p, type = "h", legend.position = "bottom")
mdaplotg(p, type = "b", show.legend = FALSE)
```



```
par(mfrow = c(2, 2))
mdaplotg(p, type = "l", show.labels = TRUE)
mdaplotg(p, type = "b", show.labels = TRUE, labels = "indices")
mdaplotg(p, type = "h", show.labels = TRUE, labels = "values")
mdaplotg(p, type = "b", show.labels = TRUE, labels = "values")
```

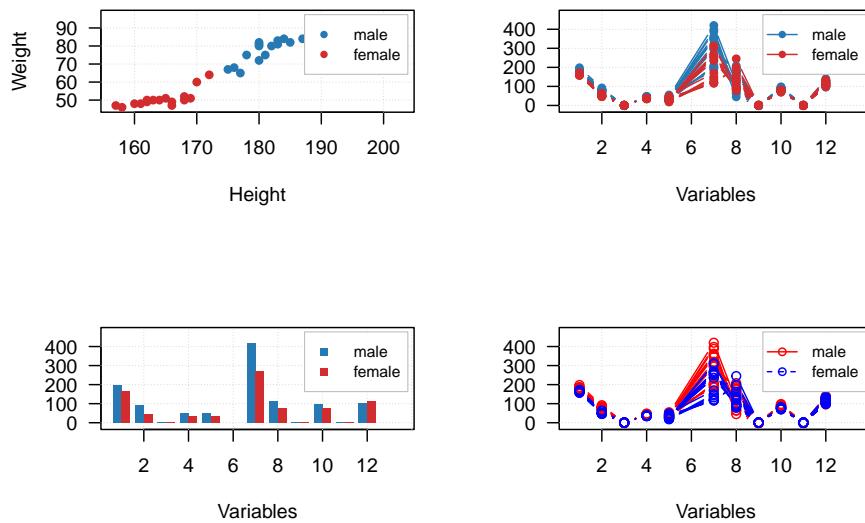


```
# load data and exclude column with income
data(people)
people = mda.exclcols(people, "Income")

# use values of sex variable to split data into two subsets
sex = people[, "Sex"]
m = mda.subset(people, subset = sex == -1)
f = mda.subset(people, subset = sex == 1)

# combine the two subsets into a named list
d = list(male = m, female = f)

# make plots for the list
par(mfrow = c(2, 2))
mdaplotg(d, type = "p")
mdaplotg(d, type = "b")
mdaplotg(d, type = "h")
mdaplotg(d, type = "b", lty = c(1, 2), col = c("red", "blue"), pch = 1)
```

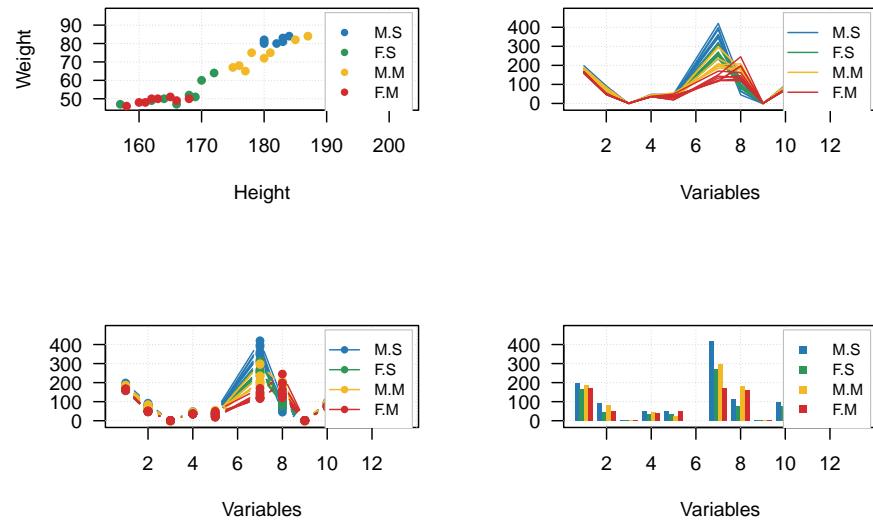


```

sex = factor(people[, "Sex"], labels = c("M", "F"))
reg = factor(people[, "Region"], labels = c("S", "M"))
groups = data.frame(sex, reg)

par(mfrow = c(2, 2))
mdaplotg(people, type = "p", groupby = groups)
mdaplotg(people, type = "l", groupby = groups)
mdaplotg(people, type = "b", groupby = groups)
mdaplotg(people, type = "h", groupby = groups)

```



Apêndices

Sites para consulta

<https://www.geeksforgeeks.org/break-axis-of-plot-in-r/> <https://stackoverflow.com/questions/72498597/add-a-break-to-a-bar-chart-when-one-value-is-very-large>

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executá-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

Referências

Chapter 11

R Modulo 4.1 - Estrutura da comunidade

RESUMO

Na ecologia de comunidades, a diversidade de espécies é uma medida importante para entender a complexidade e a estrutura de uma comunidade de organismos. Vamos explorar algumas das métricas comuns usadas para avaliar a diversidade de espécies.

Apresentação

Na ecologia de comunidades, a diversidade de espécies é uma medida importante para entender a complexidade e a estrutura de uma comunidade de organismos. Vamos explorar algumas das métricas comuns usadas para avaliar a diversidade de espécies. Essas métricas são usadas para avaliar a diversidade e a estrutura de comunidades ecológicas. Elas podem fornecer informações importantes sobre como as diferentes espécies interagem em um ecossistema e como a diversidade de espécies pode ser afetada por mudanças ambientais ou distúrbios (MAGUR-RAN, 1988).

11.1 Sobre os dados

Usaremos para esse tutorial dois conjuntos de dados. A **Matriz comunitária** (ppbio06c-peixes.xlsx) de dados coletados no Programa de Pesquisa em Biodiversidade - PPBio (Veja Programa de Pesquisa em Biodiversidade – PPBio). Esses são dados de espécies de peixes distribuídas em diversas unidades amostrais (UA's ou sítios). Essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados para os valores de Captura Por Unidade de Esforço (CPUE), nem foram relativizados ou transformados.

Além disso usaremos a **tabela de agrupamentos** (ppbio06-grupos)

Revise as informações sobre as bases de dados no Capítulo 3. A matriz de dados para esse Módulo pode ser baixada na Seção 3.2.

11.2 Organização básica

```
dev.off() #apaga os graficos, se houver algum
rm(list=ls(all=TRUE)) #limpa a memoria
cat("\014") #limpa o console
```

Instalando os pacotes necessários para esse módulo.

```
install.packages("vegan")
install.packages("moments")
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyverse")
install.packages("tidyverse") #atente para alguma msg de erro qdo executar essa linha
install.packages("forcats")
install.packages("iNEXT")
install.packages("openxlsx")
install.packages("gt")
```

Depois de instalados, carregue os pacotes a seguir no seu computador.

```
library(tibble); library(tidyverse); library(forcats); library(openxlsx); library(Rcpp)
```

Os códigos acima, são usados para instalar e carregar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades do R (R CRAN, (TEAM, 2017)) e RStudio (TEAM, 2022)). No exemplo acima, o pacote `openxlsx` permite ler e escrever arquivos Excel no R. Para instalar um pacote no R, você precisa usar a função `install.packages()`.

Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`. Por exemplo, para carregar o pacote `openxlsx`, você precisa executar a função `library(openxlsx)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando

`setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

11.3 Importando a planilha

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações.

- Ajuste a primeira linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.
- Ajuste o parâmetro `sheet = "Sheet1"` para refletir a aba correta do arquivo .xlsx a ser importado.

```
#dir <- getwd() #criamos um vetor com o diretório de trabalho
#shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06c-peixes.xlsx",
                    rowNames = T, colNames = T,
                    sheet = "Sheet1")
str(ppbio)
#View(ppbio)
ppbio[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
```

```
## 'data.frame':   26 obs. of  35 variables:
## $ ap-davis : num  0 0 0 0 0 0 0 0 0 ...
## $ as-bimac : num  1 99 194 19 23 142 5 46 206 16 ...
## $ as-fasci : num  0 0 55 0 1 3 1 0 64 0 ...
## $ ch-bimac : num  0 0 0 0 13 3 0 178 0 0 ...
## $ ci-ocela : num  0 0 0 0 0 0 40 0 0 13 ...
## $ ci-orien : num  0 0 5 0 0 69 9 0 25 24 ...
## $ co-macro : num  0 0 0 0 0 0 0 0 0 0 ...
## $ co-heter : num  0 0 1 0 0 0 0 0 0 0 ...
## $ cr-menez : num  0 0 14 0 0 4 0 0 8 0 ...
## $ cu-lepid : num  0 0 0 0 0 0 0 0 0 0 ...
## $ cy-gilbe : num  0 0 0 0 0 0 0 0 0 0 ...
## $ ge-brasi : num  0 0 3 0 0 0 0 0 1 0 ...
## $ he-margi : num  0 0 0 0 0 1 0 0 0 0 ...
## $ ho-malab : num  0 0 1 5 0 17 10 2 31 4 ...
## $ hy-pusar : num  0 0 9 2 0 43 2 0 11 0 ...
## $ le-melan : num  0 0 0 0 0 0 0 0 0 0 ...
## $ le-piau : num  0 0 3 0 0 1 3 0 2 1 ...
```

```

## $ le-taeni : num 0 0 0 0 0 0 0 0 0 ...
## $ mo-costa : num 0 0 0 0 0 0 0 0 0 ...
## $ mo-lepid : num 0 1 39 0 0 1 0 0 0 ...
## $ or-nilot : num 0 2 36 0 0 77 0 0 138 0 ...
## $ pa-manag : num 0 0 0 0 0 0 0 0 0 ...
## $ pimel-sp : num 0 0 6 0 0 0 0 0 0 ...
## $ po-retic : num 0 0 0 0 0 20 0 0 5 0 ...
## $ po-vivip : num 0 0 47 15 0 221 32 0 326 10 ...
## $ pr-brevi : num 9 0 5 0 1 15 5 2 164 0 ...
## $ ps-rhomb : num 0 0 0 0 0 0 0 1 0 ...
## $ ps-genise: num 0 0 0 0 0 0 0 1 0 ...
## $ se-heter : num 0 0 40 14 4 60 0 0 38 0 ...
## $ se-piaba : num 0 0 68 0 0 0 0 0 0 ...
## $ se-spilo : num 0 0 0 0 0 0 0 1 0 ...
## $ st-noton : num 0 0 1 0 0 25 0 0 115 0 ...
## $ sy-marmo : num 0 0 0 0 0 1 0 0 0 ...
## $ te-chalc : num 0 0 0 0 0 0 0 0 0 ...
## $ tr-signa : num 0 0 18 0 0 15 0 0 7 0 ...
##      ap-davis as-bimac as-fasci ch-bimac ci-ocela
## S-A-ZA1      0      1      0      0      0
## S-R-CC1      0     99      0      0      0
## S-R-CT1      0    194     55      0      0
## S-R-CP1      0     19      0      0      0
## S-A-TA1      0     23      1     13      0

```

Exibindo os dados importados (esses comando são “case-sensitive” `ignore.case(object)`).

```

#View(ppbio)
print(ppbio[1:8,1:8])
ppbio[1:10,1:10]
str(ppbio)
mode(ppbio)
class(ppbio)

```

11.4 Reset point

```

m_trab <- ppbio
#pat <- "S"
#m_trab <- m_trab[!grepl(pat, rownames(m_trab)), ] #exclui quem começa com pat

```

Aqui cria-se um novo objeto do R (`m_trab`, ou a matriz de trabalho, para esse momento) que substitui a matriz de dados original, por uma nova matriz que pode ser a matriz relativizada, transformada, transposta, etc. Dessa forma, mantemos a matriz de dados original caso precisemos dela novamente (Veja a

11.5. TRANSPOR A MATRIZ PARA TRABALHAR COM AS ESPÉCIES201

Tabela ??).

Revise Seção 3.2.2 e a Tabela de Abreviações (@ref(tab:200m_2)) na mesma Seção que resumem os tipos de matrizes e suas abreviações, para os nossos códigos.

11.5 Transpor a matriz para trabalhar com as espécies

Vamos transpor a matriz para trabalharmos com as espécies. A função `t` transpõe a matriz. Só deve ser usada uma vez, pois se repetida com `Ctrl+Enter` continua “girando” a matriz. As espécies como colunas representam uma matriz comunitária e as espécies como linhas representam uma matriz (comunitária) transposta.

```
m_trab <- t(ppbio)
str(m_trab)
#View(m_trab)
m_trab
print(m_trab[1:5,1:5])
m_trab[1:5,1:5]
str(m_trab)
mode(m_trab)
class(m_trab)

## num [1:35, 1:26] 0 1 0 0 0 0 0 0 0 0 ...
## - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:35] "ap-davis" "as-bimac" "as-fasci" "ch-bimac" ...
##   ..$ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
##           S-A-ZA1 S-R-CC1 S-R-CT1 S-R-CP1 S-A-TA1 S-R-CT2 S-R-CP2 S-A-TA2
## ap-davis      0      0      0      0      0      0      0      0
## as-bimac     1     99    194     19     23    142      5     46
## as-fasci     0      0     55      0      1      3      1      0
## ch-bimac     0      0      0      0     13      3      0    178
## ci-ocela     0      0      0      0      0      0     40      0
## ci-orien     0      0      5      0      0     69      9      0
## co-macro     0      0      0      0      0      0      0      0
## co-heter     0      0      1      0      0      0      0      0
## cr-menez     0      0     14      0      0      4      0      0
## cu-lepid     0      0      0      0      0      0      0      0
## cy-gilbe     0      0      0      0      0      0      0      0
## ge-brasi     0      0      3      0      0      0      0      0
## he-margi     0      0      0      0      0      1      0      0
## ho-malab     0      0      1      5      0     17     10      2
## hy-pusar     0      0      9      2      0     43      2      0
```

## le-melan	0	0	0	0	0	0	0	0
## le-piau	0	0	3	0	0	1	3	0
## le-taeni	0	0	0	0	0	0	0	0
## mo-costa	0	0	0	0	0	0	0	0
## mo-lepid	0	1	39	0	0	1	0	0
## or-nilot	0	2	36	0	0	77	0	0
## pa-manag	0	0	0	0	0	0	0	0
## pimel-sp	0	0	6	0	0	0	0	0
## po-retic	0	0	0	0	0	20	0	0
## po-vivip	0	0	47	15	0	221	32	0
## pr-brevi	9	0	5	0	1	15	5	2
## ps-rhomb	0	0	0	0	0	0	0	0
## ps-genise	0	0	0	0	0	0	0	0
## se-heter	0	0	40	14	4	60	0	0
## se-piaba	0	0	68	0	0	0	0	0
## se-spilo	0	0	0	0	0	0	0	0
## st-noton	0	0	1	0	0	25	0	0
## sy-marmo	0	0	0	0	0	0	1	0
## te-chalc	0	0	0	0	0	0	0	0
## tr-signa	0	0	18	0	0	15	0	0
	S-R-CT3	S-R-CP3	S-A-TA3	S-R-CT4	S-R-CP4	S-A-TA4	B-A-MU1	B-R-ET1
## ap-davis	0	0	0	0	0	0	0	0
## as-bimac	206	16	234	0	0	394	12	3
## as-fasci	64	0	7	1	0	0	0	0
## ch-bimac	0	0	238	0	0	273	0	0
## ci-ocela	0	13	0	0	11	0	0	0
## ci-orien	25	24	0	5	6	0	0	0
## co-macro	0	0	2	0	0	0	0	0
## co-heter	0	0	0	0	0	0	0	0
## cr-menez	8	0	0	1	0	1	0	0
## cu-lepid	0	0	0	0	0	0	0	0
## cy-gilbe	0	0	0	50	0	0	0	0
## ge-brasi	1	0	0	3	0	1	190	0
## he-margi	0	0	0	1	0	0	0	0
## ho-malab	31	4	20	4	2	9	0	0
## hy-pusar	11	0	0	3	0	0	0	0
## le-melan	0	0	0	0	0	0	0	0
## le-piau	2	1	0	0	2	2	0	0
## le-taeni	0	0	0	0	0	0	0	0
## mo-costa	0	0	0	0	0	0	0	0
## mo-lepid	0	0	0	0	0	0	0	0
## or-nilot	138	0	0	73	0	1	6	8
## pa-manag	0	0	0	0	0	0	0	1
## pimel-sp	0	0	0	0	0	0	0	0
## po-retic	5	0	0	0	0	0	0	34
## po-vivip	326	10	0	28	80	0	0	0

11.5. TRANSPOR A MATRIZ PARA TRABALHAR COM AS ESPÉCIES203

## pr-brevi	164	0	0	59	0	3	0	0	
## ps-rhomb	1	0	0	0	0	0	0	0	
## ps-genise	1	0	0	0	0	0	0	0	
## se-heter	38	0	0	3	3	0	0	0	
## se-piaba	0	0	0	0	0	0	0	0	
## se-spilo	1	0	0	0	0	0	0	0	
## st-noton	115	0	0	64	0	0	0	0	
## sy-marmo	0	0	0	0	0	0	0	0	
## te-chalc	0	0	0	0	0	0	0	0	
## tr-signa	7	0	0	141	0	0	0	0	
		B-A-GU1	B-R-PC2	B-A-MU2	B-A-GU2	B-R-PC3	B-A-MU3	B-A-GU3	B-R-PC4
## ap-davis	0	5	0	0	22	0	0	0	0
## as-bimac	2	44	99	0	75	511	6	7	
## as-fasci	2	0	0	0	7	0	0	17	
## ch-bimac	0	0	0	0	0	0	0	0	
## ci-ocela	0	2	0	0	4	0	0	0	
## ci-orien	0	0	0	0	0	0	0	0	
## co-macro	0	0	0	0	0	0	0	0	
## co-heter	0	0	0	0	0	0	0	0	
## cr-menez	0	0	0	0	0	0	0	0	
## cu-lepid	0	0	0	0	21	0	0	0	
## cy-gilbe	0	0	0	0	0	0	0	81	
## ge-brasi	7	8	67	23	16	145	32	5	
## he-margi	0	0	0	0	0	0	0	0	
## ho-malab	0	0	1	0	2	0	0	1	
## hy-pusar	0	0	0	0	1	0	0	0	
## le-melan	0	2	0	0	0	0	0	0	
## le-piau	0	0	0	0	0	0	0	1	
## le-taeni	0	1	0	0	0	0	0	0	
## mo-costa	0	0	0	0	1	0	0	0	
## mo-lepid	0	0	0	0	0	0	0	0	
## or-nilot	3	5	1	36	65	11	247	9	
## pa-manag	11	0	0	102	0	0	250	0	
## pimel-sp	0	0	0	0	0	0	0	0	
## po-retic	0	0	10	0	0	46	0	0	
## po-vivip	0	0	8	0	0	48	0	0	
## pr-brevi	0	9	0	0	6	1	0	0	
## ps-rhomb	0	0	0	0	0	0	0	0	
## ps-genise	0	0	0	0	0	0	0	0	
## se-heter	0	10	0	0	93	0	0	31	
## se-piaba	0	0	0	0	0	0	0	0	
## se-spilo	0	0	0	0	0	0	0	0	
## st-noton	0	0	0	0	0	0	0	0	
## sy-marmo	0	0	0	0	0	0	0	0	
## te-chalc	0	76	0	0	58	0	0	0	
## tr-signa	0	23	0	0	0	0	0	4	

```

##          B-A-MU4 B-A-GU4
## ap-davis      0      0
## as-bimac    235     13
## as-fasci      0      0
## ch-bimac      0      0
## ci-ocela      0      0
## ci-orien      0      0
## co-macro      0      0
## co-heter      0      0
## cr-menez      0      0
## cu-lepid      0      0
## cy-gilbe      0      0
## ge-brasi     509     10
## he-margi      0      0
## ho-malab      0      0
## hy-pusar      0      0
## le-melan      0      0
## le-piau       0      0
## le-taeni      0      0
## mo-costa      0      0
## mo-lepid      0      0
## or-nilot       1    129
## pa-manag      0    190
## pimel-sp       0      0
## po-retic     266      0
## po-vivip     163      0
## pr-brevi       0      0
## ps-rhomb      0      0
## ps-genise      0      0
## se-heter       0      0
## se-piaba      0      0
## se-spilo       0      0
## st-noton      0      0
## sy-marmo      0      0
## te-chalc       0      0
## tr-signa      0      0
##          S-A-ZA1 S-R-CC1 S-R-CT1 S-R-CP1 S-A-TA1
## ap-davis      0      0      0      0      0
## as-bimac      1     99    194     19     23
## as-fasci      0      0     55      0      1
## ch-bimac      0      0      0      0     13
## ci-ocela      0      0      0      0      0
##          S-A-ZA1 S-R-CC1 S-R-CT1 S-R-CP1 S-A-TA1
## ap-davis      0      0      0      0      0
## as-bimac      1     99    194     19     23
## as-fasci      0      0     55      0      1

```

11.5. TRANSPOR A MATRIZ PARA TRABALHAR COM AS ESPÉCIES205

```
## ch-bimac      0      0      0      0     13
## ci-ocela      0      0      0      0      0
## num [1:35, 1:26] 0 1 0 0 0 0 0 0 0 ...
## - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:35] "ap-davis" "as-bimac" "as-fasci" "ch-bimac" ...
##   ..$ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
## [1] "numeric"
## [1] "matrix" "array"
```

11.5.1 Informações básicas da matriz

Agora podemos pedir ao R as informações básicas da matriz de trabalho (`m_trab`), como o número de observações ou tamanho do vetor (depende do tipo da matriz), número de observações igual a zero, número de observações maiores que zero e proporção de zeros na matriz.

```
range(m_trab) #menor e maior valores
length(m_trab) #no. de colunas
ncol(m_trab) #no. de N colunas
nrow(m_trab) #no. de M linhas
sum(lengths(m_trab)) #soma os nos. de colunas
length(as.matrix(m_trab)) #tamanho da matriz m x n
sum(m_trab == 0) #número de observações igual a zero
sum(m_trab > 0) #número de observações maiores que zero
#calculando a proporção de zeros na matriz
zeros <- (sum(m_trab == 0)/length(as.matrix(m_trab)))*100
zeros
```

```
## [1] 0 511
## [1] 910
## [1] 26
## [1] 35
## [1] 910
## [1] 910
## [1] 716
## [1] 194
## [1] 78.68132
```

Tabela que resume as informações geradas (Tabela 11.1).

##	Comando	Resultado
## 1	range	0 - 511
## 2	length	910
## 3	n_cols	26
## 4	m_linhas	35
## 5	Tamanho	910
## 6	Tamanho	910

```
## 7      Zeros      716
## 8 Nao zeros     194
## 9 % Zeros      78.7
```

Table 11.1: Resumo das informações sobre o tamanho da base de dados.

Comando	Resultado
range	0 - 511
length	910
n_cols	26
m_linhas	35
Tamanho	910
Tamanho	910
Zeros	716
Nao zeros	194
% Zeros	78.7

Ou seja, temos uma matriz de tamanho $m \times n$ igual a 35 objetos por 26 atributos, onde 78.68% dos valores da matriz são iguais a zero!

Agora de conhecimento dessas informações básicas podemos calcular os primeiros descritores da estrutura da comunidade a ser estudada.

11.6 Calculando os descritores da comunidade

Entre outras métricas, calcularemos os seguintes índices:

1. Riqueza de Espécies:

- A riqueza de espécies simplesmente se refere ao número total de espécies diferentes em uma comunidade. É uma medida fundamental da diversidade ecológica e reflete a variedade de formas de vida coexistentes em um ecossistema. Comunidades com alta riqueza de espécies têm um grande número de espécies diferentes, enquanto comunidades com baixa riqueza têm menos espécies.

2. Índice de Diversidade de Simpson:

- O índice de diversidade de Simpson (ou índice de Simpson) mede a probabilidade de escolher aleatoriamente duas vezes o mesmo indivíduo de uma comunidade. Quanto mais próximo de 1 for o índice de Simpson, menor é a diversidade, indicando que uma ou algumas espécies dominam a comunidade. Quanto mais próximo de 0 for o índice de Simpson, maior é a diversidade, indicando uma comunidade mais equilibrada.

3. Índice de Diversidade de Shannon-Wiener:

- O índice de Shannon-Wiener (ou índice de Shannon) leva em consideração a riqueza de espécies e a equitabilidade (distribuição uniforme das abundâncias das espécies). Ele mede a incerteza associada à identificação de uma espécie aleatória em uma comunidade. Quanto maior o índice de Shannon, maior é a diversidade, pois indica uma comunidade com várias espécies bem distribuídas em termos de abundância.

4. Equitabilidade:

- A equitabilidade é uma medida que avalia o quanto uniformemente as abundâncias das diferentes espécies estão distribuídas em uma comunidade. Quanto maior a equitabilidade, mais igual é a distribuição das abundâncias, o que indica uma comunidade mais equilibrada.

5. Abundância:

- A abundância se refere ao número total de indivíduos de uma espécie em uma comunidade. É uma medida simples que indica quantos indivíduos de uma espécie específica estão presentes na comunidade.

6. Abundância Relativa:

- A abundância relativa é a proporção ou a fração da abundância de uma espécie em relação à abundância total de todas as espécies na comunidade. É uma medida que ajuda a entender a importância relativa de cada espécie na comunidade.

7. Dominância de Espécies:

- A dominância de espécies se refere à presença de uma ou algumas espécies que têm uma abundância significativamente maior do que as outras na comunidade. Comunidades com alta dominância são frequentemente menos diversas, pois algumas espécies dominantes podem suprimir o crescimento de outras.

11.6.1 Variabilidade

Primeiro a variabilidade estatística

```
#?apply
Sum <- rowSums(m_trab)
#ou
Sum <- apply(m_trab, 1, sum)
Sum
## Abundância relativa (%)
RA <- (Sum / sum(Sum)) * 100 # percentage
## Media
```

```

Mean <- rowMeans(m_trab)
Mean
## Ou
Mean <- apply(m_trab, 1, mean)
Mean
## Desvio padrão
DP <- apply(m_trab, 1, sd)
DP
## Máximo
Max <- apply(m_trab, 1, max)
Max
## Mínimo
Min <- apply(m_trab, 1, min)
Min
## Mínimo não-zero
MinZ <- apply(m_trab, 1, function(row) {
  non_zero_values <- row[row > 0] # Filter out zero values
  if (length(non_zero_values) == 0) {
    return(0) # If all values are zero, return 0
  } else {
    return(min(non_zero_values)) # Return the minimum of non-zero values
  }
})
MinZ

##   ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien   co-macro   co-heter
##      27        2386       158        705        70       143         2          1
##   cr-menez   cu-lepid   cy-gilbe   ge-brasi   he-margi   ho-malab   hy-pusar   le-melan
##      28        21        131       1020        2        109        71          2
##   le-piau   le-taeni   mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
##      15        1         1        41        848       554         6        381
##   po-vivip   pr-brevi   ps-rhomb   ps-genise   se-heter   se-piaba   se-spilo   st-noton
##      978       279        1         1       296        68         1       205
##   sy-marmo   te-chalc   tr-signa
##      1        134       208
##   ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien
## 1.03846154 91.76923077 6.07692308 27.11538462 2.69230769 5.50000000
##   co-macro   co-heter   cr-menez   cu-lepid   cy-gilbe   ge-brasi
## 0.07692308 0.03846154 1.07692308 0.80769231 5.03846154 39.23076923
##   he-margi   ho-malab   hy-pusar   le-melan   le-piau   le-taeni
## 0.07692308 4.19230769 2.73076923 0.07692308 0.57692308 0.03846154
##   mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
## 0.03846154 1.57692308 32.61538462 21.30769231 0.23076923 14.65384615
##   po-vivip   pr-brevi   ps-rhomb   ps-genise   se-heter   se-piaba
## 37.61538462 10.73076923 0.03846154 0.03846154 11.38461538 2.61538462

```

```

##    se-spilo   st-noton   sy-marmo   te-chalc   tr-signa
## 0.03846154  7.88461538 0.03846154  5.15384615 8.00000000
##    ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien
## 1.03846154 91.76923077 6.07692308 27.11538462 2.69230769 5.50000000
##    co-macro   co-heter   cr-menez   cu-lepid   cy-gilbe   ge-brasi
## 0.07692308 0.03846154 1.07692308 0.80769231 5.03846154 39.23076923
##    he-margi   ho-malab   hy-pusar   le-melan   le-piau   le-taeni
## 0.07692308 4.19230769 2.73076923 0.07692308 0.57692308 0.03846154
##    mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
## 0.03846154 1.57692308 32.61538462 21.30769231 0.23076923 14.65384615
##    po-vivip   pr-brevi   ps-rhomb   ps-genise   se-heter   se-piaba
## 37.61538462 10.73076923 0.03846154 0.03846154 11.38461538 2.61538462
##    se-spilo   st-noton   sy-marmo   te-chalc   tr-signa
## 0.03846154  7.88461538 0.03846154  5.15384615 8.00000000
##    ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien
## 4.3861671 132.4348316 16.2035134 75.8732242 8.2982853 14.6184815
##    co-macro   co-heter   cr-menez   cu-lepid   cy-gilbe   ge-brasi
## 0.3922323 0.1961161 3.1486261 4.1184388 18.3313519 106.3478473
##    he-margi   ho-malab   hy-pusar   le-melan   le-piau   le-taeni
## 0.2717465 7.6053625 8.6605205 0.3922323 0.9868364 0.1961161
##    mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
## 0.1961161 7.6376597 59.1452970 62.3928004 1.1766968 52.5242362
##    po-vivip   pr-brevi   ps-rhomb   ps-genise   se-heter   se-piaba
## 79.5004790 33.3892889 0.1961161 0.1961161 22.9905666 13.3358972
##    se-spilo   st-noton   sy-marmo   te-chalc   tr-signa
## 0.1961161 25.5582893 0.1961161 18.3841068 27.8280434
##    ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien   co-macro   co-heter
##      22       511       64       273       40       69       2       1
##    cr-menez   cu-lepid   cy-gilbe   ge-brasi   he-margi   ho-malab   hy-pusar   le-melan
##      14       21       81       509       1       31       43       2
##    le-piau   le-taeni   mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
##      3        1        1        39       247      250       6     266
##    po-vivip   pr-brevi   ps-rhomb   ps-genise   se-heter   se-piaba   se-spilo   st-noton
##      326      164       1       1       93       68       1     115
##    sy-marmo   te-chalc   tr-signa
##      1       76      141
##    ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien   co-macro   co-heter
##      0        0        0        0        0        0        0        0
##    cr-menez   cu-lepid   cy-gilbe   ge-brasi   he-margi   ho-malab   hy-pusar   le-melan
##      0        0        0        0        0        0        0        0
##    le-piau   le-taeni   mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
##      0        0        0        0        0        0        0        0
##    po-vivip   pr-brevi   ps-rhomb   ps-genise   se-heter   se-piaba   se-spilo   st-noton
##      0        0        0        0        0        0        0        0
##    sy-marmo   te-chalc   tr-signa
##      0        0        0

```

```

##  ap-davis  as-bimac  as-fasci  ch-bimac  ci-ocela  ci-orien  co-macro  co-heter
##      5          1          1          3          2          5          2          1
##  cr-menez  cu-lepid  cy-gilbe  ge-brasi  he-margi  ho-malab  hy-pusar  le-melan
##      1         21         50          1          1          1          1          2
##  le-piau   le-taeni  mo-costa  mo-lepid  or-nilot  pa-manag  pimel-sp  po-retic
##      1          1          1          1          1          1          6          5
##  po-vivip  pr-brevi  ps-rhomb  ps-genise  se-heter  se-piaba  se-spilo  st-noton
##      8          1          1          1          3         68          1          1
##  sy-marmo  te-chalc  tr-signa
##      1         58          4

```

11.6.2 Riqueza

Atente para o fato de que a riqueza será a frequêcia de ocorrência na matrix transposta. Converte-se primeiro para matriz binária.

```

m_pa <- m_trab
m_pa[m_pa != 0] <- 1
rowSums(m_pa)
library(vegan)
bin <- decostand(m_trab, "pa")
bin[1:10, 1:10]
S <- apply(bin, 1, sum)
S
#OU
Riqueza <- specnumber(m_trab)
Riqueza
Riqueza_total <- specnumber(colSums(m_trab))
Riqueza_total
#OU
FO <- rowSums(m_trab > 0) / ncol(m_trab) * 100
FO

##  ap-davis  as-bimac  as-fasci  ch-bimac  ci-ocela  ci-orien  co-macro  co-heter
##      2         23         10          5          5          7          1          1
##  cr-menez  cu-lepid  cy-gilbe  ge-brasi  he-margi  ho-malab  hy-pusar  le-melan
##      5          1          2         15          2         14          7          1
##  le-piau   le-taeni  mo-costa  mo-lepid  or-nilot  pa-manag  pimel-sp  po-retic
##      8          1          1          3         18          5          1          6
##  po-vivip  pr-brevi  ps-rhomb  ps-genise  se-heter  se-piaba  se-spilo  st-noton
##     11         12          1          1         10          1          1          4
##  sy-marmo  te-chalc  tr-signa
##      1          2          6
##          S-A-ZA1  S-R-CC1  S-R-CT1  S-R-CP1  S-A-TA1  S-R-CT2  S-R-CP2  S-A-TA2
##  ap-davis      0          0          0          0          0          0          0          0
##  as-bimac      1          1          1          1          1          1          1          1

```

```

## as-fasci      0      0      1      0      1      1      1      0
## ch-bimac     0      0      0      0      1      1      0      1
## ci-ocela     0      0      0      0      0      0      1      0
## ci-orien     0      0      1      0      0      1      1      0
## co-macro     0      0      0      0      0      0      0      0
## co-heter     0      0      1      0      0      0      0      0
## cr-menez     0      0      1      0      0      1      0      0
## cu-lepid     0      0      0      0      0      0      0      0
##           S-R-CT3 S-R-CP3
## ap-davis     0      0
## as-bimac     1      1
## as-fasci     1      0
## ch-bimac     0      0
## ci-ocela     0      1
## ci-orien     1      1
## co-macro     0      0
## co-heter     0      0
## cr-menez     1      0
## cu-lepid     0      0
## ap-davis   as-bimac  as-fasci  ch-bimac  ci-ocela  ci-orien  co-macro  co-heter
##      2       23      10       5       5       7       1       1
## cr-menez   cu-lepid  cy-gilbe  ge-brasi  he-margi  ho-malab  hy-pusar  le-melan
##      5       1       2      15       2      14       7       1
## le-piau    le-taeni  mo-costa mo-lepid  or-nilot  pa-manag  pimel-sp  po-retic
##      8       1       1       3      18       5       1       6
## po-vivip   pr-brevi ps-rhomb ps-genise se-heter  se-piaba  se-spilo  st-noton
##      11      12      1       1      10       1       1       4
## sy-marmo  te-chalc tr-signa
##      1       2       6
## ap-davis   as-bimac  as-fasci  ch-bimac  ci-ocela  ci-orien  co-macro  co-heter
##      2       23      10       5       5       7       1       1
## cr-menez   cu-lepid  cy-gilbe  ge-brasi  he-margi  ho-malab  hy-pusar  le-melan
##      5       1       2      15       2      14       7       1
## le-piau    le-taeni  mo-costa mo-lepid  or-nilot  pa-manag  pimel-sp  po-retic
##      8       1       1       3      18       5       1       6
## po-vivip   pr-brevi ps-rhomb ps-genise se-heter  se-piaba  se-spilo  st-noton
##      11      12      1       1      10       1       1       4
## sy-marmo  te-chalc tr-signa
##      1       2       6
## [1] 26
## ap-davis   as-bimac  as-fasci  ch-bimac  ci-ocela  ci-orien  co-macro  co-heter
## 7.692308 88.461538 38.461538 19.230769 19.230769 26.923077 3.846154 3.846154
## cr-menez   cu-lepid  cy-gilbe  ge-brasi  he-margi  ho-malab  hy-pusar  le-melan
## 19.230769 3.846154 7.692308 57.692308 7.692308 53.846154 26.923077 3.846154
## le-piau    le-taeni  mo-costa mo-lepid  or-nilot  pa-manag  pimel-sp  po-retic
## 30.769231 3.846154 3.846154 11.538462 69.230769 19.230769 3.846154 23.076923

```

```
## po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton
## 42.307692 46.153846 3.846154 3.846154 38.461538 3.846154 3.846154 15.384615
## sy-marmo te-chalc tr-signa
## 3.846154 7.692308 23.076923
```

11.7 Índices de Diversidade

11.7.1 Shannon

O índice de diversidade de Shannon ou Shannon–Weaver (ou Shannon–Wiener) é uma medida de diversidade que leva em consideração tanto a riqueza de espécies quanto a uniformidade na distribuição dessas espécies (DIXON, 2003; OKSANEN et al., 2020). Este índice é definido por:

$$H' = - \sum_i p_i \log_b p_i$$

Onde:

- H' é a entropia de Shannon.
- p_i é a abundância proporcional da espécie i .
- b é a base do logaritmo.

É mais comum usar-se o logaritmo natural, embora pode-se argumentar que para o logaritmo de base = 2 (o que faz sentido mas nenhuma diferença)

Outra fórmula para calcular o índice de diversidade de Shannon é:

$$H' = - \sum_{i=1}^S \left(\frac{n_i}{N} \times \ln \frac{n_i}{N} \right)$$

Onde:

- H' é a entropia de Shannon (ou a diversidade de Shannon).
- n_i é o número de indivíduos da espécie i .
- N é o número total de indivíduos na comunidade.
- S é o número total de espécies na comunidade.

Esta fórmula mede a incerteza (ou entropia) na identificação de uma espécie selecionada aleatoriamente da comunidade. Quanto maior o valor de H' , maior a diversidade da comunidade.

```
H <- diversity(m_trab, index = "shannon")
H
```

```
## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
## 0.4791656 2.4239775 1.4761308 1.1783625 1.1883832 1.4976803 0.0000000 0.0000000
## cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab hy-pusar le-melan
```

```

## 1.2205076 0.0000000 0.6648803 1.5668772 0.6931472 2.1167591 1.2492728 0.0000000
## le-piau le-taeni mo-costa mo-lepid or-nilot pa-manag pimel-sp po-retic
## 1.9913464 0.0000000 0.0000000 0.2287207 2.1072815 1.1268689 0.0000000 1.0288585
## po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton
## 1.8633952 1.3708151 0.0000000 0.0000000 1.8675926 0.0000000 0.0000000 0.9702899
## sy-marmo te-chalc tr-signa
## 0.0000000 0.6840978 1.0985693

```

11.7.2 Simpson

O índice de diversidade de Simpson é uma medida de diversidade que leva em consideração a riqueza de espécies e a abundância relativa de cada espécie em uma comunidade. A fórmula para calcular o índice de diversidade de Simpson é:

$$D = 1 - \sum_{i=1}^S \left(\frac{n_i(n_i - 1)}{N(N - 1)} \right)$$

Onde:

- D é o índice de diversidade de Simpson.
- n_i é o número de indivíduos da espécie i .
- N é o número total de indivíduos na comunidade.
- S é o número total de espécies na comunidade.

Esta fórmula fornece um valor entre 0 e 1, onde 0 indica uma comunidade com apenas uma espécie presente e 1 indica uma comunidade com uma distribuição uniforme de espécies.

Existem duas variantes do índice de Simpson baseado em $D = \sum p_i^2$ (DIXON, 2003; HURLBERT, 1971; OKSANEN et al., 2020):

- Ao escolhermos `simpson` o R retorna $1 - D$, e
- ao escolhermos `invsimpson` o R retorna $1/D$.

```

D <- diversity(m_trab, "simpson")
D
D[is.na(D)] <- 0 #substitui NA ou NaN por 0
D

```

```

## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro
## 0.30178326 0.88451845 0.69860599 0.67197827 0.61020408 0.70027874 0.00000000
## co-heter cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab
## 0.00000000 0.64540816 0.00000000 0.47200047 0.68977124 0.50000000 0.83982830
## hy-pusar le-melan le-piau le-taeni mo-costa mo-lepid or-nilot
## 0.58956556 0.00000000 0.85333333 0.00000000 0.00000000 0.09399167 0.83992302
## pa-manag pimel-sp po-retic po-vivip pr-brevi ps-rhomb ps-genise

```

```

## 0.64444343 0.00000000 0.48641164 0.79634160 0.60348659 0.00000000 0.00000000
## se-heter se-piaba se-spilo st-noton sy-marmo te-chalc tr-signa
## 0.81071950 0.00000000 0.00000000 0.57294468 0.00000000 0.49097795 0.51405325
## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro
## 0.30178326 0.88451845 0.69860599 0.67197827 0.61020408 0.70027874 0.00000000
## co-heter cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab
## 0.00000000 0.64540816 0.00000000 0.47200047 0.68977124 0.50000000 0.83982830
## hy-pusar le-melan le-piau le-taeni mo-costa mo-lepid or-nilot
## 0.58956556 0.00000000 0.85333333 0.00000000 0.00000000 0.09399167 0.83992302
## pa-manag pimel-sp po-retic po-vivip pr-brevi ps-rhomb ps-genise
## 0.64444343 0.00000000 0.48641164 0.79634160 0.60348659 0.00000000 0.00000000
## se-heter se-piaba se-spilo st-noton sy-marmo te-chalc tr-signa
## 0.81071950 0.00000000 0.00000000 0.57294468 0.00000000 0.49097795 0.51405325

```

11.7.3 Equitabilidade de Pielou

O índice de equitabilidade de Pielou é uma medida de uniformidade em uma distribuição de espécies (PIELOU, 1975). Sua fórmula é dada por:

$$J' = \frac{H'}{\ln(S)}$$

Onde:

- J' é o índice de equitabilidade de Pielou.
- H' é a entropia de Shannon (ou a diversidade de Shannon), que é calculada como $-\sum_{i=1}^S p_i \cdot \ln(p_i)$, onde S é o número total de espécies e p_i é a proporção da espécie i .
- $\ln(S)$ é o logarítmico natural do número total de espécies na comunidade.

Esta fórmula fornece um valor entre 0 e 1, onde 0 indica uma distribuição completamente desigual onde uma comunidade apresenta uma única espécie dominante e 1 indica uma distribuição completamente equitativa onde todas as espécies têm a mesma abundância.

```

E <- H/log(specnumber(m_trab))
E
E[is.na(E)] <- 0 #substitui NA ou NaN por 0
E

## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
## 0.6912899 0.7730767 0.6410755 0.7321578 0.7383840 0.7696554      NaN      NaN
## cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab hy-pusar le-melan
## 0.7583440      NaN 0.9592195 0.5785997 1.0000000 0.8020891 0.6419992      NaN
## le-piau le-taeni mo-costa mo-lepid or-nilot pa-manag pimel-sp po-retic
## 0.9576352      NaN      NaN 0.2081906 0.7290694 0.7001630      NaN 0.5742169

```

```

## po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton
## 0.7770962 0.5516566      NaN      NaN 0.8110852      NaN      NaN 0.6999162
## sy-marmo te-chalc tr-signa
##      NaN 0.9869445 0.6131232
## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
## 0.6912899 0.7730767 0.6410755 0.7321578 0.7383840 0.7696554 0.0000000 0.0000000
## cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab hy-pusar le-melan
## 0.7583440 0.0000000 0.9592195 0.5785997 1.0000000 0.8020891 0.6419992 0.0000000
## le-piau le-taeni mo-costa mo-lepid or-nilot pa-manag pimel-sp po-retic
## 0.9576352 0.0000000 0.0000000 0.2081906 0.7290694 0.7001630 0.0000000 0.5742169
## po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton
## 0.7770962 0.5516566 0.0000000 0.0000000 0.8110852 0.0000000 0.0000000 0.6999162
## sy-marmo te-chalc tr-signa
## 0.0000000 0.9869445 0.6131232

```

11.7.3.1 Assimetria e curtose

A assimetria mede a falta de simetria em uma distribuição de frequência. Uma distribuição é simétrica se as duas metades à esquerda e à direita da média são cópias espelhadas uma da outra. Se a distribuição não é simétrica, então é assimétrica. A assimetria pode ser positiva, negativa ou nula (simétrica). Uma assimetria positiva indica que a cauda da distribuição se estende mais para a direita em relação à média, enquanto uma assimetria negativa indica que a cauda da distribuição se estende mais para a esquerda em relação à média.

A curtose descreve o pico ou a “pontiagudez” de uma distribuição. Uma distribuição com alta curtose tem uma alta concentração de valores ao redor da média e caudas mais pesadas (ou seja, valores extremos são mais prováveis). Uma distribuição com baixa curtose é mais achatada e dispersa, com caudas mais leves. A curtose pode ser positiva (distribuição leptocúrtica, com alta concentração em torno da média e caudas pesadas), negativa (distribuição platocúrtica, com dispersão alta e caudas mais leves) ou nula (mesocúrtica, similar à distribuição normal).

Essas medidas são úteis para compreender as propriedades e características de diferentes conjuntos de dados e distribuições de frequência. Elas ajudam a compreender a forma e o comportamento dos dados em uma amostra ou população (ZAR, 1999).

```

library(moments)
Assimetria <- apply(m_trab,1,skewness)
Assimetria
Curtose <- apply(m_trab,1,kurtosis)
Curtose

## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
## 4.469425 1.774738 2.959549 2.544388 3.796856 3.482201 4.800000 4.800000

```

```

##  cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab hy-pusar le-melan
## 3.252387 4.800000 3.484076 3.638552 3.175426 2.282120 4.127075 4.800000
##  le-piau le-taeni mo-costa mo-lepid or-nilot pa-manag pimel-sp po-retic
## 1.437653 4.800000 4.800000 4.790304 2.287297 2.867520 4.800000 4.462822
##  po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton
## 2.544468 4.046241 4.800000 4.800000 2.293389 4.800000 4.800000 3.419722
##  sy-marmo te-chalc tr-signa
## 4.800000 3.281874 4.442741
##  ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
## 21.730097 5.556931 10.273303 7.736436 17.146010 15.065210 24.040000 24.040000
##  cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab hy-pusar le-melan
## 12.820351 24.040000 13.776840 16.059988 11.083333 7.604321 19.499005 24.040000
##  le-piau le-taeni mo-costa mo-lepid or-nilot pa-manag pimel-sp po-retic
## 3.655586 24.040000 24.040000 23.979582 7.972989 9.860393 24.040000 21.825541
##  po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton
## 8.674340 18.659676 24.040000 24.040000 7.676958 24.040000 24.040000 13.817921
##  sy-marmo te-chalc tr-signa
## 24.040000 12.021173 21.715543

```

11.7.3.2 Tabela de descritores

Muito confuso? Criamos na sequência uma tabela final com todos os descritores da comunidade e da normalidade.

11.7.3.2.1 Descritores da estrutura da comunidade: Espécies Sum, soma; RA, abundância relativa (%), mean, média; DP, desvio padrão da média; Max, maior valor; Min, menor valor; S, riqueza (ou frequência de ocorrência na matriz transposta); E, índice de equitabilidade de Pielou (PIELOU, 1975); H, índice de diversidade de Shannon (LUDWIG; REYNOLDS, 1988); D, índice de diversidade de Simpson (HURLBERT, 1971).

```

Descritores1 <- cbind(Sum, RA, Mean, DP, Max, Min, MinZ, FO, S, E, H, D)
Descritores1 <- as.data.frame(Descritores1)
Descritores1
#Descritores1 <- Descritores1 %>% rownames_to_column(var="Espécies") #da nome a primeira coluna
SomaTotalD <- apply(Descritores1,2,sum)
SomaTotalD
MediaTotalD <- apply(Descritores1,2,mean)
MediaTotalD
DPTotalD <- apply(Descritores1,2,sd)
DPTotalD
Descritores2 <- cbind(SomaTotalD, MediaTotalD, DPTotalD)
Descritores2 <- as.data.frame(Descritores2)
Descritores2 <- t(Descritores2)
Descritores2

```

```
DescritoresFinal <- rbind(Descritores1, Descritores2)
DescritoresFinal
DescritoresFinal <- round (DescritoresFinal, 2)
DescritoresFinal
```

	Sum	RA	Mean	DP	Max	Min	MinZ	F0	S
## ap-davis	27	0.30354132	1.03846154	4.3861671	22	0	5	7.692308	2
## as-bimac	2386	26.82405846	91.76923077	132.4348316	511	0	1	88.461538	23
## as-fasci	158	1.77627881	6.07692308	16.2035134	64	0	1	38.461538	10
## ch-bimac	705	7.92580101	27.11538462	75.8732242	273	0	3	19.230769	5
## ci-ocela	70	0.78695897	2.69230769	8.2982853	40	0	2	19.230769	5
## ci-orien	143	1.60764474	5.50000000	14.6184815	69	0	5	26.923077	7
## co-macro	2	0.02248454	0.07692308	0.3922323	2	0	2	3.846154	1
## co-heter	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## cr-menez	28	0.31478359	1.07692308	3.1486261	14	0	1	19.230769	5
## cu-lepid	21	0.23608769	0.80769231	4.1184388	21	0	21	3.846154	1
## cy-gilbe	131	1.47273749	5.03846154	18.3313519	81	0	50	7.692308	2
## ge-brasi	1020	11.46711636	39.23076923	106.3478473	509	0	1	57.692308	15
## he-margi	2	0.02248454	0.07692308	0.2717465	1	0	1	7.692308	2
## ho-malab	109	1.22540753	4.19230769	7.6053625	31	0	1	53.846154	14
## hy-pusar	71	0.79820124	2.73076923	8.6605205	43	0	1	26.923077	7
## le-melan	2	0.02248454	0.07692308	0.3922323	2	0	2	3.846154	1
## le-piau	15	0.16863406	0.57692308	0.9868364	3	0	1	30.769231	8
## le-taeni	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## mo-costa	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## mo-lepid	41	0.46093311	1.57692308	7.6376597	39	0	1	11.538462	3
## or-nilot	848	9.53344576	32.61538462	59.1452970	247	0	1	69.230769	18
## pa-manag	554	6.22821810	21.30769231	62.3928004	250	0	1	19.230769	5
## pimel-sp	6	0.06745363	0.23076923	1.1766968	6	0	6	3.846154	1
## po-retic	381	4.28330523	14.65384615	52.5242362	266	0	5	23.076923	6
## po-vivip	978	10.99494098	37.61538462	79.5004790	326	0	8	42.307692	11
## pr-brevi	279	3.13659359	10.73076923	33.3892889	164	0	1	46.153846	12
## ps-rhomb	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## ps-genise	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## se-heter	296	3.32771220	11.38461538	22.9905666	93	0	3	38.461538	10
## se-piaba	68	0.76447442	2.61538462	13.3358972	68	0	68	3.846154	1
## se-spilo	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## st-noton	205	2.30466554	7.88461538	25.5582893	115	0	1	15.384615	4
## sy-marmo	1	0.01124227	0.03846154	0.1961161	1	0	1	3.846154	1
## te-chalc	134	1.50646431	5.15384615	18.3841068	76	0	58	7.692308	2
## tr-signa	208	2.33839236	8.00000000	27.8280434	141	0	4	23.076923	6
		E	H	D					
## ap-davis	0.6912899	0.4791656	0.30178326						
## as-bimac	0.7730767	2.4239775	0.88451845						
## as-fasci	0.6410755	1.4761308	0.69860599						

```

## ch-bimac  0.7321578 1.1783625 0.67197827
## ci-ocela  0.7383840 1.1883832 0.61020408
## ci-orien  0.7696554 1.4976803 0.70027874
## co-macro  0.0000000 0.0000000 0.00000000
## co-heter  0.0000000 0.0000000 0.00000000
## cr-menez  0.7583440 1.2205076 0.64540816
## cu-lepid  0.0000000 0.0000000 0.00000000
## cy-gilbe  0.9592195 0.6648803 0.47200047
## ge-brasi  0.5785997 1.5668772 0.68977124
## he-margi  1.0000000 0.6931472 0.50000000
## ho-malab  0.8020891 2.1167591 0.83982830
## hy-pusar  0.6419992 1.2492728 0.58956556
## le-melan  0.0000000 0.0000000 0.00000000
## le-piau   0.9576352 1.9913464 0.85333333
## le-taeni  0.0000000 0.0000000 0.00000000
## mo-costa  0.0000000 0.0000000 0.00000000
## mo-lepid  0.2081906 0.2287207 0.09399167
## or-nilot  0.7290694 2.1072815 0.83992302
## pa-manag  0.7001630 1.1268689 0.64444343
## pimel-sp  0.0000000 0.0000000 0.00000000
## po-retic  0.5742169 1.0288585 0.48641164
## po-vivip  0.7770962 1.8633952 0.79634160
## pr-brevi  0.5516566 1.3708151 0.60348659
## ps-rhomb  0.0000000 0.0000000 0.00000000
## ps-genise 0.0000000 0.0000000 0.00000000
## se-heter  0.8110852 1.8675926 0.81071950
## se-piaba  0.0000000 0.0000000 0.00000000
## se-spilo  0.0000000 0.0000000 0.00000000
## st-noton  0.6999162 0.9702899 0.57294468
## sy-marmo  0.0000000 0.0000000 0.00000000
## te-chalc  0.9869445 0.6840978 0.49097795
## tr-signa  0.6131232 1.0985693 0.51405325
##           Sum      RA      Mean      DP      Max      Min      MinZ
## 8895.00000 100.00000 342.11538 807.30587 3484.00000 0.00000 262.00000
##          F0       S       E       H       D
## 746.15385 194.00000 16.69499 30.09298 14.31057
##           Sum      RA      Mean      DP      Max      Min
## 254.1428571 2.8571429 9.7747253 23.0658821 99.5428571 0.0000000
##          MinZ      F0       S       E       H       D
## 7.4857143 21.3186813 5.5428571 0.4769997 0.8597994 0.4088734
##           Sum      RA      Mean      DP      Max      Min
## 468.3865386 5.2657284 18.0148669 33.0609233 138.2715450 0.0000000
##          MinZ      F0       S       E       H       D
## 16.4573807 21.3807831 5.5590036 0.3757324 0.7784814 0.3354621
##           Sum      RA      Mean      DP      Max Min      MinZ
## SomaTotalD 8895.0000 100.000000 342.115385 807.30587 3484.00000 0 262.00000

```

```

## MediaTotalD 254.1429  2.857143  9.774725 23.06588 99.54286  0 7.485714
## DPTotalD   468.3865  5.265728 18.014867 33.06092 138.27155  0 16.457381
##          F0      S      E      H      D
## SomaTotalD 746.15385 194.000000 16.6949878 30.0929803 14.3105692
## MediaTotalD 21.31868  5.542857  0.4769997 0.8597994 0.4088734
## DPTotalD   21.38078  5.559004  0.3757324 0.7784814 0.3354621
##          Sum     RA    Mean     DP      Max Min
## ap-davis    27.0000 0.30354132 1.03846154 4.3861671 22.00000 0
## as-bimac   2386.0000 26.82405846 91.76923077 132.4348316 511.00000 0
## as-fasci    158.0000 1.77627881 6.07692308 16.2035134 64.00000 0
## ch-bimac    705.0000 7.92580101 27.11538462 75.8732242 273.00000 0
## ci-ocela    70.0000 0.78695897 2.69230769 8.2982853 40.00000 0
## ci-orien   143.0000 1.60764474 5.50000000 14.6184815 69.00000 0
## co-macro    2.0000 0.02248454 0.07692308 0.3922323 2.00000 0
## co-heter    1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## cr-menez   28.0000 0.31478359 1.07692308 3.1486261 14.00000 0
## cu-lepid   21.0000 0.23608769 0.80769231 4.1184388 21.00000 0
## cy-gilbe   131.0000 1.47273749 5.03846154 18.3313519 81.00000 0
## ge-brasi   1020.0000 11.46711636 39.23076923 106.3478473 509.00000 0
## he-margi    2.0000 0.02248454 0.07692308 0.2717465 1.00000 0
## ho-malab   109.0000 1.22540753 4.19230769 7.6053625 31.00000 0
## hy-pusar    71.0000 0.79820124 2.73076923 8.6605205 43.00000 0
## le-melan    2.0000 0.02248454 0.07692308 0.3922323 2.00000 0
## le-piau    15.0000 0.16863406 0.57692308 0.9868364 3.00000 0
## le-taeni   1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## mo-costa   1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## mo-lepid   41.0000 0.46093311 1.57692308 7.6376597 39.00000 0
## or-nilot   848.0000 9.53344576 32.61538462 59.1452970 247.00000 0
## pa-manag   554.0000 6.22821810 21.30769231 62.3928004 250.00000 0
## pimel-sp    6.0000 0.06745363 0.23076923 1.1766968 6.00000 0
## po-retic   381.0000 4.28330523 14.65384615 52.5242362 266.00000 0
## po-vivip   978.0000 10.99494098 37.61538462 79.5004790 326.00000 0
## pr-brevi   279.0000 3.13659359 10.73076923 33.3892889 164.00000 0
## ps-rhomb   1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## ps-genise   1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## se-heter   296.0000 3.32771220 11.38461538 22.9905666 93.00000 0
## se-piaba   68.0000 0.76447442 2.61538462 13.3358972 68.00000 0
## se-spilo    1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## st-noton   205.0000 2.30466554 7.88461538 25.5582893 115.00000 0
## sy-marmo   1.0000 0.01124227 0.03846154 0.1961161 1.00000 0
## te-chalc   134.0000 1.50646431 5.15384615 18.3841068 76.00000 0
## tr-signa   208.0000 2.33839236 8.00000000 27.8280434 141.00000 0
## SomaTotalD 8895.0000 100.00000000 342.11538462 807.3058718 3484.00000 0
## MediaTotalD 254.1429  2.85714286  9.77472527 23.0658821 99.54286  0
## DPTotalD   468.3865  5.26572837 18.01486687 33.0609233 138.27155  0
##          MinZ     F0      S      E      H      D

```

```

## ap-davis      5.000000  7.692308  2.000000  0.6912899  0.4791656  0.30178326
## as-bimac     1.000000  88.461538 23.000000  0.7730767  2.4239775  0.88451845
## as-fasci     1.000000  38.461538 10.000000  0.6410755  1.4761308  0.69860599
## ch-bimac     3.000000  19.230769  5.000000  0.7321578  1.1783625  0.67197827
## ci-ocela     2.000000  19.230769  5.000000  0.7383840  1.1883832  0.61020408
## ci-orien     5.000000  26.923077  7.000000  0.7696554  1.4976803  0.70027874
## co-macro     2.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## co-heter     1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## cr-menez     1.000000  19.230769  5.000000  0.7583440  1.2205076  0.64540816
## cu-lepid    21.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## cy-gilbe    50.000000  7.692308  2.000000  0.9592195  0.6648803  0.47200047
## ge-brasi     1.000000  57.692308 15.000000  0.5785997  1.5668772  0.68977124
## he-margi     1.000000  7.692308  2.000000  1.0000000  0.6931472  0.50000000
## ho-malab     1.000000  53.846154 14.000000  0.8020891  2.1167591  0.83982830
## hy-pusar     1.000000  26.923077  7.000000  0.6419992  1.2492728  0.58956556
## le-melan     2.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## le-piau      1.000000  30.769231  8.000000  0.9576352  1.9913464  0.85333333
## le-taeni     1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## mo-costa     1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## mo-lepid     1.000000  11.538462  3.000000  0.2081906  0.2287207  0.09399167
## or-nilot     1.000000  69.230769 18.000000  0.7290694  2.1072815  0.83992302
## pa-manag     1.000000  19.230769  5.000000  0.7001630  1.1268689  0.64444343
## pimel-sp     6.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## po-retic     5.000000  23.076923  6.000000  0.5742169  1.0288585  0.48641164
## po-vivip     8.000000  42.307692 11.000000  0.7770962  1.8633952  0.79634160
## pr-brevi     1.000000  46.153846 12.000000  0.5516566  1.3708151  0.60348659
## ps-rhomb     1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## ps-genise    1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## se-heter     3.000000  38.461538 10.000000  0.8110852  1.8675926  0.81071950
## se-piaba    68.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## se-spilo     1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## st-noton     1.000000  15.384615  4.000000  0.6999162  0.9702899  0.57294468
## sy-marmo     1.000000  3.846154   1.000000  0.0000000  0.0000000  0.00000000
## te-chalc    58.000000  7.692308  2.000000  0.9869445  0.6840978  0.49097795
## tr-signa     4.000000  23.076923  6.000000  0.6131232  1.0985693  0.51405325
## SomaTotalD  262.000000 746.153846 194.000000 16.6949878 30.0929803 14.31056920
## MediaTotalD  7.485714  21.318681  5.542857  0.4769997  0.8597994  0.40887341
## DPTotalD   16.457381  21.380783  5.559004  0.3757324  0.7784814  0.33546207
##          Sum   RA   Mean     DP   Max Min  MinZ   FO     S     E
## ap-davis    27.00  0.30  1.04   4.39  22.00  0  5.00  7.69  2.00  0.69
## as-bimac   2386.00 26.82 91.77 132.43 511.00  0  1.00  88.46 23.00 0.77
## as-fasci    158.00  1.78  6.08  16.20  64.00  0  1.00  38.46 10.00 0.64
## ch-bimac    705.00  7.93 27.12  75.87 273.00  0  3.00 19.23  5.00 0.73
## ci-ocela     70.00  0.79  2.69  8.30  40.00  0  2.00 19.23  5.00 0.74
## ci-orien    143.00  1.61  5.50 14.62  69.00  0  5.00 26.92  7.00 0.77
## co-macro     2.00  0.02  0.08  0.39  2.00  0  2.00  3.85  1.00 0.00

```



```

## hy-pusar    1.25  0.59
## le-melan   0.00  0.00
## le-piau    1.99  0.85
## le-taeni   0.00  0.00
## mo-costa   0.00  0.00
## mo-lepid   0.23  0.09
## or-nilot    2.11  0.84
## pa-manag   1.13  0.64
## pimel-sp    0.00  0.00
## po-retic   1.03  0.49
## po-vivip   1.86  0.80
## pr-brevi    1.37  0.60
## ps-rhomb   0.00  0.00
## ps-genise   0.00  0.00
## se-heter    1.87  0.81
## se-piaba   0.00  0.00
## se-spilo    0.00  0.00
## st-noton   0.97  0.57
## sy-marmo   0.00  0.00
## te-chalc   0.68  0.49
## tr-signa   1.10  0.51
## SomaTotalD 30.09 14.31
## MediaTotalD 0.86  0.41
## DPTotalD   0.78  0.34

```

```

#Fazendo uma tabela
library(gt)
df <- DescritoresFinal
ncol(df); nrow(df) #no. de N colunas x M linhas
df <- cbind(Spp = rownames(df), df)
gt(df, rowname_col = "Espécie", caption = "Descritores da diversidade por espécie (colu

```

```

## [1] 12
## [1] 38

```

Spp	Sum	RA	Mean	DP	Max	Min	MinZ	FO	S	F
ap-davis	27.00	0.30	1.04	4.39	22.00	0	5.00	7.69	2.00	0.68
as-bimac	2386.00	26.82	91.77	132.43	511.00	0	1.00	88.46	23.00	0.77
as-fasci	158.00	1.78	6.08	16.20	64.00	0	1.00	38.46	10.00	0.64
ch-bimac	705.00	7.93	27.12	75.87	273.00	0	3.00	19.23	5.00	0.73
ci-ocela	70.00	0.79	2.69	8.30	40.00	0	2.00	19.23	5.00	0.74
ci-orien	143.00	1.61	5.50	14.62	69.00	0	5.00	26.92	7.00	0.77
co-macro	2.00	0.02	0.08	0.39	2.00	0	2.00	3.85	1.00	0.00
co-heter	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00
cr-menez	28.00	0.31	1.08	3.15	14.00	0	1.00	19.23	5.00	0.76

cu-lepid	21.00	0.24	0.81	4.12	21.00	0	21.00	3.85	1.00	0.00	0.00
cy-gilbe	131.00	1.47	5.04	18.33	81.00	0	50.00	7.69	2.00	0.96	0.66
ge-brasi	1020.00	11.47	39.23	106.35	509.00	0	1.00	57.69	15.00	0.58	1.57
he-margi	2.00	0.02	0.08	0.27	1.00	0	1.00	7.69	2.00	1.00	0.69
ho-malab	109.00	1.23	4.19	7.61	31.00	0	1.00	53.85	14.00	0.80	2.12
hy-pusar	71.00	0.80	2.73	8.66	43.00	0	1.00	26.92	7.00	0.64	1.25
le-melan	2.00	0.02	0.08	0.39	2.00	0	2.00	3.85	1.00	0.00	0.00
le-piau	15.00	0.17	0.58	0.99	3.00	0	1.00	30.77	8.00	0.96	1.99
le-taeni	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00	0.00
mo-costa	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00	0.00
mo-lepid	41.00	0.46	1.58	7.64	39.00	0	1.00	11.54	3.00	0.21	0.23
or-nilot	848.00	9.53	32.62	59.15	247.00	0	1.00	69.23	18.00	0.73	2.11
pa-manag	554.00	6.23	21.31	62.39	250.00	0	1.00	19.23	5.00	0.70	1.13
pimel-sp	6.00	0.07	0.23	1.18	6.00	0	6.00	3.85	1.00	0.00	0.00
po-retic	381.00	4.28	14.65	52.52	266.00	0	5.00	23.08	6.00	0.57	1.03
po-vivip	978.00	10.99	37.62	79.50	326.00	0	8.00	42.31	11.00	0.78	1.86
pr-brevi	279.00	3.14	10.73	33.39	164.00	0	1.00	46.15	12.00	0.55	1.37
ps-rhomb	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00	0.00
ps-genise	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00	0.00
se-heter	296.00	3.33	11.38	22.99	93.00	0	3.00	38.46	10.00	0.81	1.87
se-piaba	68.00	0.76	2.62	13.34	68.00	0	68.00	3.85	1.00	0.00	0.00
se-spilo	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00	0.00
st-noton	205.00	2.30	7.88	25.56	115.00	0	1.00	15.38	4.00	0.70	0.97
sy-marmo	1.00	0.01	0.04	0.20	1.00	0	1.00	3.85	1.00	0.00	0.00
te-chalc	134.00	1.51	5.15	18.38	76.00	0	58.00	7.69	2.00	0.99	0.68
tr-signa	208.00	2.34	8.00	27.83	141.00	0	4.00	23.08	6.00	0.61	1.10
SomaTotalD	8895.00	100.00	342.12	807.31	3484.00	0	262.00	746.15	194.00	16.69	30.09
MediaTotalD	254.14	2.86	9.77	23.07	99.54	0	7.49	21.32	5.54	0.48	0.86
DPTotalD	468.39	5.27	18.01	33.06	138.27	0	16.46	21.38	5.56	0.38	0.78

11.7.3.2.2 Descritores da normalidade Agora tabelamos os descritores de normalidade

```
Normalidade1 <- cbind(Assimetria, Curtose)
Normalidade1 <- as.data.frame(Normalidade1)
Normalidade1
SomaTotalN <- apply(Normalidade1, 2, sum)
SomaTotalN
MediaTotalN <- apply(Normalidade1, 2, mean)
MediaTotalN
DPTotalN <- apply(Normalidade1, 2, sd)
DPTotalN
Normalidade2<-cbind(SomaTotalN, MediaTotalN, DPTotalN)
Normalidade2<-as.data.frame(Normalidade2)
```

```

Normalidade2 <- t(Normalidade2) "t" transpõe a matriz
Normalidade2
NormalidadeFinal <- rbind(Normalidade1, Normalidade2)
NormalidadeFinal
NormalidadeFinal <- round(NormalidadeFinal, 2)
NormalidadeFinal

```

	Assimetria	Curtose
## ap-davis	4.469425	21.730097
## as-bimac	1.774738	5.556931
## as-fasci	2.959549	10.273303
## ch-bimac	2.544388	7.736436
## ci-ocela	3.796856	17.146010
## ci-orien	3.482201	15.065210
## co-macro	4.800000	24.040000
## co-heter	4.800000	24.040000
## cr-menez	3.252387	12.820351
## cu-lepid	4.800000	24.040000
## cy-gilbe	3.484076	13.776840
## ge-brasi	3.638552	16.059988
## he-margi	3.175426	11.083333
## ho-malab	2.282120	7.604321
## hy-pusar	4.127075	19.499005
## le-melan	4.800000	24.040000
## le-piau	1.437653	3.655586
## le-taeni	4.800000	24.040000
## mo-costa	4.800000	24.040000
## mo-lepid	4.790304	23.979582
## or-nilot	2.287297	7.972989
## pa-manag	2.867520	9.860393
## pimel-sp	4.800000	24.040000
## po-retic	4.462822	21.825541
## po-vivip	2.544468	8.674340
## pr-brevi	4.046241	18.659676
## ps-rhomb	4.800000	24.040000
## ps-genise	4.800000	24.040000
## se-heter	2.293389	7.676958
## se-piaba	4.800000	24.040000
## se-spilo	4.800000	24.040000
## st-noton	3.419722	13.817921
## sy-marmo	4.800000	24.040000
## te-chalc	3.281874	12.021173
## tr-signa	4.442741	21.715543
## Assimetria	Curtose	
## 132.4608	596.6915	

```
## Assimetria    Curtose
## 3.784595 17.048329
## Assimetria    Curtose
## 1.043802 6.944634
##           Assimetria    Curtose
## SomaTotalN 132.460824 596.691528
## MediaTotalN 3.784595 17.048329
## DPTotalN   1.043802 6.944634
##           Assimetria    Curtose
## ap-davis     4.469425 21.730097
## as-bimac     1.774738 5.556931
## as-fasci     2.959549 10.273303
## ch-bimac     2.544388 7.736436
## ci-ocela     3.796856 17.146010
## ci-orien     3.482201 15.065210
## co-macro     4.800000 24.040000
## co-heter     4.800000 24.040000
## cr-menez     3.252387 12.820351
## cu-lepid     4.800000 24.040000
## cy-gilbe     3.484076 13.776840
## ge-brasi     3.638552 16.059988
## he-margi     3.175426 11.083333
## ho-malab     2.282120 7.604321
## hy-pusar     4.127075 19.499005
## le-melan     4.800000 24.040000
## le-piau      1.437653 3.655586
## le-taeni     4.800000 24.040000
## mo-costa     4.800000 24.040000
## mo-lepid     4.790304 23.979582
## or-nilot     2.287297 7.972989
## pa-manag     2.867520 9.860393
## pimel-sp     4.800000 24.040000
## po-retic     4.462822 21.825541
## po-vivip     2.544468 8.674340
## pr-brevi      4.046241 18.659676
## ps-rhomb     4.800000 24.040000
## ps-genise    4.800000 24.040000
## se-heter     2.293389 7.676958
## se-piaba     4.800000 24.040000
## se-spilo     4.800000 24.040000
## st-noton     3.419722 13.817921
## sy-marmo     4.800000 24.040000
## te-chalc     3.281874 12.021173
## tr-signa     4.442741 21.715543
## SomaTotalN 132.460824 596.691528
## MediaTotalN 3.784595 17.048329
```

```

## DPTotalN      1.043802   6.944634
##               Assimetria Curtose
## ap-davis      4.47    21.73
## as-bimac     1.77    5.56
## as-fasci     2.96    10.27
## ch-bimac     2.54    7.74
## ci-ocela      3.80    17.15
## ci-orien      3.48    15.07
## co-macro      4.80    24.04
## co-heter      4.80    24.04
## cr-menez     3.25    12.82
## cu-lepid      4.80    24.04
## cy-gilbe      3.48    13.78
## ge-brasi      3.64    16.06
## he-margi      3.18    11.08
## ho-malab      2.28    7.60
## hy-pusar      4.13    19.50
## le-melan      4.80    24.04
## le-piau       1.44    3.66
## le-taeni      4.80    24.04
## mo-costa      4.80    24.04
## mo-lepid      4.79    23.98
## or-nilot      2.29    7.97
## pa-manag      2.87    9.86
## pimel-sp      4.80    24.04
## po-retic      4.46    21.83
## po-vivip      2.54    8.67
## pr-brevi       4.05    18.66
## ps-rhomb      4.80    24.04
## ps-genise     4.80    24.04
## se-heter      2.29    7.68
## se-piaba      4.80    24.04
## se-spilo      4.80    24.04
## st-noton      3.42    13.82
## sy-marmo      4.80    24.04
## te-chalc      3.28    12.02
## tr-signa      4.44    21.72
## SomaTotalN    132.46   596.69
## MediaTotalN   3.78    17.05
## DPTotalN      1.04    6.94

```

A função `fix(nome da matriz)` dá acesso ao grid da matriz criada para manipulação dos dados numéricos.

```
#Fazendo uma tabela
nf <- NormalidadeFinal
ncol(nf); nrow(nf) #no. de N colunas x M linhas
nf <- cbind(Spp = rownames(nf), nf)
gt(nf, rowname_col = "Espécie", caption = "Descritores da normalidade por espécie (coluna)")

## [1] 2
## [1] 38
```

Spp	Assimetria	Curtose
ap-davis	4.47	21.73
as-bimac	1.77	5.56
as-fasci	2.96	10.27
ch-bimac	2.54	7.74
ci-ocela	3.80	17.15
ci-orien	3.48	15.07
co-macro	4.80	24.04
co-heter	4.80	24.04
cr-menez	3.25	12.82
cu-lepid	4.80	24.04
cy-gilbe	3.48	13.78
ge-brasi	3.64	16.06
he-margi	3.18	11.08
ho-malab	2.28	7.60
hy-pusar	4.13	19.50
le-melan	4.80	24.04
le-piau	1.44	3.66
le-taeni	4.80	24.04
mo-costa	4.80	24.04
mo-lepid	4.79	23.98
or-nilot	2.29	7.97
pa-manag	2.87	9.86
pimel-sp	4.80	24.04
po-retic	4.46	21.83
po-vivip	2.54	8.67
pr-brevi	4.05	18.66
ps-rhomb	4.80	24.04
ps-genise	4.80	24.04
se-heter	2.29	7.68
se-piaba	4.80	24.04
se-spiolo	4.80	24.04
st-noton	3.42	13.82
sy-marmo	4.80	24.04
te-chalc	3.28	12.02

tr-signa	4.44	21.72
SomaTotalN	132.46	596.69
MediaTotalN	3.78	17.05
DPTotalN	1.04	6.94

Nos Apêndices você pode ver o script para os mesmos descritores para as Unidades Amostrais (UA's)

11.7.3.3 Lidando com células vazias

Em R, `NaN` (Not a Number) e `NA` (Not Available) são valores especiais que representam dados ausentes ou valores inválidos em um vetor, matriz, data frame ou outra estrutura de dados.

- **NaN (Not a Number):** É usado para representar resultados inválidos em operações matemáticas, como a divisão por zero ou a operação de raiz quadrada de um número negativo.
- **NA (Not Available):** É usado para indicar dados ausentes. Pode ser usado em contextos onde o valor real está ausente ou desconhecido.

Em muitas operações, o `NA` é tratado como um valor especial que propagará em outras operações. Isso significa que se uma operação é realizada com um ou mais valores `NA`, o resultado geralmente será `NA`. Por outro lado, `NaN` é um valor específico que indica um resultado matematicamente indefinido ou inválido.

Por exemplo, ao realizar operações em um vetor que contém `NA`, o resultado será `NA` para qualquer operação que envolva um valor `NA`:

```
x <- c(1, 2, NA, 4)
mean(x) # Resultado será NA porque há um valor NA no vetor
```

```
## [1] NA
```

Por outro lado, `NaN` resulta de operações matemáticas inválidas:

```
0/0 # Resultado será NaN, pois a divisão por zero é indefinida
sqrt(-1) # Resultado será NaN, pois não há raiz quadrada real de um número negativo
```

```
## Warning in sqrt(-1): NaNs produzidos
```

```
## [1] NaN
## [1] NaN
```

Em resumo, enquanto `NA` indica dados ausentes ou não disponíveis, `NaN` indica resultados de operações matematicamente inválidos.

```

write.table(data.frame("Spp"=rownames(DescritoresFinal),
                      DescritoresFinal),
            "DescritoresSPP.txt",
            row.names=FALSE,
            sep="\t")
write.table(data.frame("Spp"=rownames(NormalidadeFinal),
                      NormalidadeFinal),
            "NormalidadeSPP.txt",
            row.names=FALSE,
            sep="\t")

```

11.7.3.3.1 Salvando as tabelas criadas em .txt direto no diretório de trabalho Agora é necessário voltar à matriz comunitária (**antes de ter sido transposta**), antes de continuarmos a análise, porque as análises anteriores também precisam ser refeitas com a matriz não transposta. A seguir confere-se se está sendo usada a matriz comunitária com as espécies nas colunas.

Para ver o código dos descritores da estrutura da comunidade para as Unidades Amostrais (UA's), matriz não transposta, veja Descritores da estrutura da comunidade: UA's nos apêndices.

m_trab

	S-A-ZA1	S-R-CC1	S-R-CT1	S-R-CP1	S-A-TA1	S-R-CT2	S-R-CP2	S-A-TA2
## ap-davis	0	0	0	0	0	0	0	0
## as-bimac	1	99	194	19	23	142	5	46
## as-fasci	0	0	55	0	1	3	1	0
## ch-bimac	0	0	0	0	13	3	0	178
## ci-ocela	0	0	0	0	0	0	40	0
## ci-orien	0	0	5	0	0	69	9	0
## co-macro	0	0	0	0	0	0	0	0
## co-heter	0	0	1	0	0	0	0	0
## cr-menez	0	0	14	0	0	4	0	0
## cu-lepid	0	0	0	0	0	0	0	0
## cy-gilbe	0	0	0	0	0	0	0	0
## ge-brasi	0	0	3	0	0	0	0	0
## he-margi	0	0	0	0	0	1	0	0
## ho-malab	0	0	1	5	0	17	10	2
## hy-pusar	0	0	9	2	0	43	2	0
## le-melan	0	0	0	0	0	0	0	0
## le-piau	0	0	3	0	0	1	3	0
## le-taeni	0	0	0	0	0	0	0	0
## mo-costa	0	0	0	0	0	0	0	0
## mo-lepid	0	1	39	0	0	1	0	0

## se-spilo	1	0	0	0	0	0	0	0
## st-noton	115	0	0	64	0	0	0	0
## sy-marmo	0	0	0	0	0	0	0	0
## te-chalc	0	0	0	0	0	0	0	0
## tr-signa	7	0	0	141	0	0	0	0
	B-A-GU1	B-R-PC2	B-A-MU2	B-A-GU2	B-R-PC3	B-A-MU3	B-A-GU3	B-R-PC4
## ap-davis	0	5	0	0	22	0	0	0
## as-bimac	2	44	99	0	75	511	6	7
## as-fasci	2	0	0	0	7	0	0	17
## ch-bimac	0	0	0	0	0	0	0	0
## ci-ocela	0	2	0	0	4	0	0	0
## ci-orien	0	0	0	0	0	0	0	0
## co-macro	0	0	0	0	0	0	0	0
## co-heter	0	0	0	0	0	0	0	0
## cr-menez	0	0	0	0	0	0	0	0
## cu-lepid	0	0	0	0	21	0	0	0
## cy-gilbe	0	0	0	0	0	0	0	81
## ge-brasi	7	8	67	23	16	145	32	5
## he-margi	0	0	0	0	0	0	0	0
## ho-malab	0	0	1	0	2	0	0	1
## hy-pusar	0	0	0	0	1	0	0	0
## le-melan	0	2	0	0	0	0	0	0
## le-piau	0	0	0	0	0	0	0	1
## le-taeni	0	1	0	0	0	0	0	0
## mo-costa	0	0	0	0	1	0	0	0
## mo-lepid	0	0	0	0	0	0	0	0
## or-nilot	3	5	1	36	65	11	247	9
## pa-manag	11	0	0	102	0	0	250	0
## pimel-sp	0	0	0	0	0	0	0	0
## po-retic	0	0	10	0	0	46	0	0
## po-vivip	0	0	8	0	0	48	0	0
## pr-brevi	0	9	0	0	6	1	0	0
## ps-rhomb	0	0	0	0	0	0	0	0
## ps-genise	0	0	0	0	0	0	0	0
## se-heter	0	10	0	0	93	0	0	31
## se-piaba	0	0	0	0	0	0	0	0
## se-spilo	0	0	0	0	0	0	0	0
## st-noton	0	0	0	0	0	0	0	0
## sy-marmo	0	0	0	0	0	0	0	0
## te-chalc	0	76	0	0	58	0	0	0
## tr-signa	0	23	0	0	0	0	0	4
	B-A-MU4	B-A-GU4						
## ap-davis	0	0						
## as-bimac	235	13						
## as-fasci	0	0						
## ch-bimac	0	0						

```

## ci-ocela      0      0
## ci-orien     0      0
## co-macro     0      0
## co-heter     0      0
## cr-menez     0      0
## cu-lepid     0      0
## cy-gilbe     0      0
## ge-brasi    509     10
## he-margi     0      0
## ho-malab     0      0
## hy-pusar     0      0
## le-melan     0      0
## le-piau      0      0
## le-taeni     0      0
## mo-costa     0      0
## mo-lepid     0      0
## or-nilot     1     129
## pa-manag     0     190
## pimel-sp      0      0
## po-retic    266     0
## po-vivip    163     0
## pr-brevi      0      0
## ps-rhomb     0      0
## ps-genise     0      0
## se-heter      0      0
## se-piaba     0      0
## se-spilo      0      0
## st-noton      0      0
## sy-marmo     0      0
## te-chalc      0      0
## tr-signa     0      0

```

```
m_trab <- ppbio
```

Note que em uma matriz comunitária (onde os atributos são as espécies), a matriz transposta vai conter as unidades amostrais como colunas e espécies como linhas. Não tendo a mesma interpretação de matriz transposta em matemática.

Agora faremos uma série de gráficos para a matriz comunitária.

11.8 Gráficos descritivos

11.8.1 Série de Hill e perfil da diversidade

Pode-se utilizar uma série de Hill ao invés dos índices específicos de diversidade. Assim, quanto maior o valor de q (definido em `scales`), maior será o peso para a

equabilidade. Quanto mais próximo de zero, maior o peso para riqueza. Quando $q = 0$, o valor de Hill é igual a riqueza de espécies.

```
Hill <- renyi(m_trab,
              scales = c(0:5),
              hill = TRUE)
Hill

##          0         1         2         3         4         5
## S-A-ZA1  2 1.384145 1.219512 1.170411 1.150768 1.140762
## S-R-CC1  3 1.163476 1.060983 1.045792 1.040607 1.038021
## S-R-CT1 18 8.433432 5.644824 4.479923 3.925407 3.628965
## S-R-CP1  5 4.089106 3.729963 3.562264 3.464767 3.397679
## S-A-TA1  5 2.988203 2.463687 2.265900 2.160258 2.093130
## S-R-CT2 17 8.120284 5.912807 4.995084 4.517360 4.233135
## S-R-CP2 10 5.475674 4.064111 3.571020 3.347621 3.223289
## S-A-TA2  4 1.820866 1.537624 1.437328 1.389021 1.362287
## S-R-CT3 18 7.820420 6.079509 5.349788 4.920580 4.635690
## S-R-CP3  6 4.641083 4.135957 3.852830 3.659934 3.518793
## S-A-TA3  5 2.507947 2.244026 2.186542 2.165167 2.154376
## S-R-CT4 14 6.554517 5.259698 4.665939 4.284640 4.026809
## S-R-CP4  6 2.359044 1.645269 1.479932 1.418625 1.388109
## S-A-TA4  8 2.249743 2.035378 1.981428 1.947091 1.920271
## B-A-MU1  3 1.418412 1.192503 1.145257 1.128263 1.119791
## B-R-ET1  4 2.200997 1.720325 1.562987 1.494812 1.458881
## B-A-GU1  5 3.959865 3.342246 3.016650 2.834369 2.720943
## B-R-PC2 11 5.630175 4.005266 3.427912 3.151799 2.991159
## B-A-MU2  6 2.863745 2.393193 2.248679 2.175626 2.127807
## B-A-GU2  3 2.464739 2.119634 1.930295 1.826854 1.766551
## B-R-PC3 13 7.253371 5.945359 5.447786 5.182436 5.009929
## B-A-MU3  6 2.710687 2.025352 1.799132 1.699897 1.647087
## B-A-GU3  4 2.537469 2.297723 2.232385 2.206183 2.192731
## B-R-PC4  9 4.374404 3.048096 2.586659 2.377604 2.263942
## B-A-MU4  5 3.672499 3.348321 3.103452 2.925311 2.799120
## B-A-GU4  4 2.514025 2.206452 2.107190 2.053368 2.015850
```

Podemos mapear isso em um gráfico de perfil de diversidade

```
library(ggplot2)
library(tidyr)
library(tidyverse)
grafico1 <- Hill %>%
  rownames_to_column() %>%
  pivot_longer(-rowname) %>%
  mutate(name = factor(name, name[1:length(Hill)])) %>%
  ggplot(aes(x = name, y = value, group = rowname,
```

```

        col = rowname)) +
geom_point(size = 2) +
geom_line(size = 1) +
xlab("Parâmetro de ordem de diversidade (q)") +
ylab("Diversidade") +
labs(col = "Locais") +
theme_bw() +
theme(text = element_text(size = 10)) #ajustar a fonte caso nao caiba no output html

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

grafico1
ggsave(grafico1, dpi = 300, filename = "fig-hill.png")

## Saving 6.5 x 4.5 in image

```

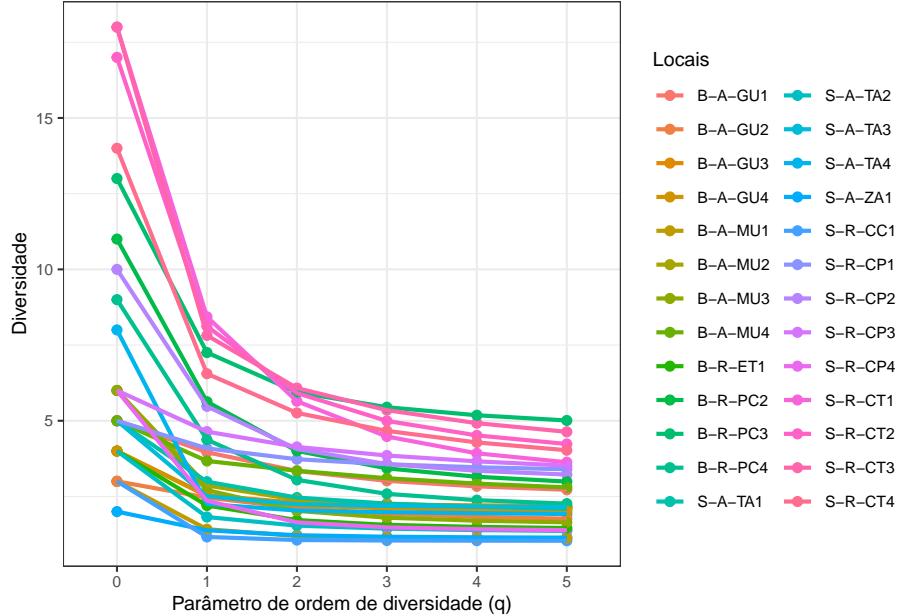


Figure 11.1: Mapeamento dos pontos de coleta em um gráfico de perfil de diversidade.

11.9 Distribuição de abundância

Uma maneira de observar a diversidade de espécies é usando um gráfico de distribuição de abundâncias de uma comunidade, como mostrado a seguir. Note que aqui utiliza-se as abundâncias totais das espécies (Figura 11.2), mas é possível fazer por linha, basta substituir o objeto `abund` pela abundância de uma linha. Como na linha marcada com o `#`.

```
abund <- colSums(m_trab)
```

Na primeira linha do código abaixo, retira-se o `#` do início da linha para rodar a distribuição de abundância apenas para o local (linha) 1 `comunidade[1,]`. Pode-se mudar o número para o local de interesse.

O código a seguir produzirá o gráfico para o local escolhido.

```
#abund <- m_trab[1, ] #escolhe a primeira linha para a distribuição de abundância
df <- data.frame(sp = colnames(m_trab),
                  abun = abund)
grafico2 <- ggplot(df, aes(fct_reorder(sp, -abun),
                           abun, group = 1)) +
  geom_col() +
  geom_line(col = "red", linetype = "dashed") +
  geom_point(col = "red") +
  xlab("Espécies") +
  ylab("Abundância") +
  theme_bw() +
  theme(axis.text.x = element_text(
    angle = 45,
    hjust = 1,
    face = "italic"))
grafico2
ggsave(grafico2, dpi = 300, filename = "fig-abun.png")
```

```
## Saving 6.5 x 4.5 in image
```

11.10 Curva de rarefação

O código abaixo indica como usar o pacote iNEXT para extrapolar e interpolar curvas de rarefação visando comparar a riqueza de diferentes locais (linhas, por isso se trabalha com a matriz transposta). A curva de rarefação vai ser construída para as UA's demarcadas em `rarefa <- t(m_trab[c("S-R-CT2", "S-R-CP2", "S-A-TA2", "B-A-MU2"),])`

Pode-se, alternativamente, construir a curva de rarefação para as 8 linhas (UA's) com maior somatório dos valores das espécies (linhas marcadas com

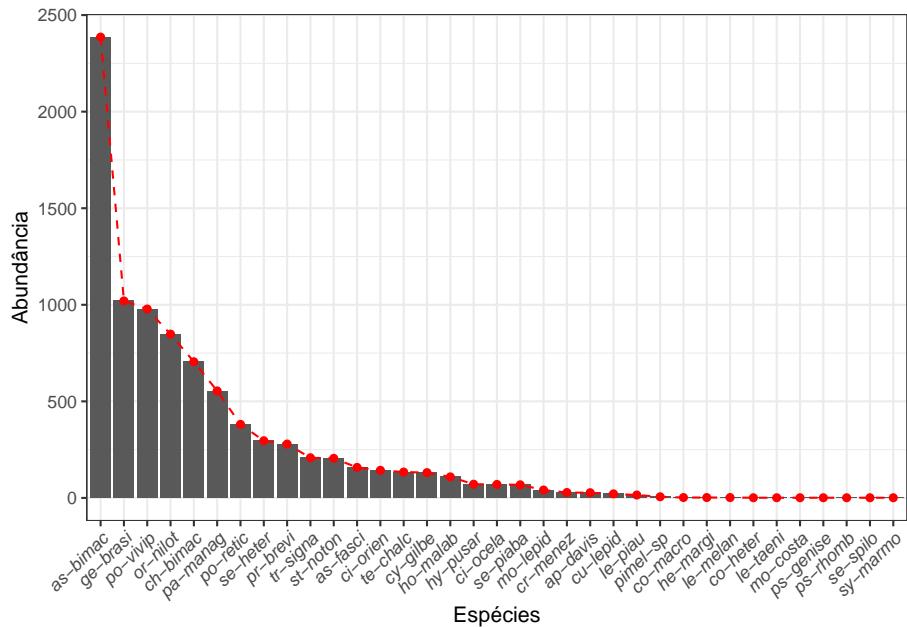


Figure 11.2: Distribuição da densidade de indivíduos.

o símbolo #).

```
m_trab_r <- mutate(m_trab, across(everything(), ceiling)) #arredonda a matriz original
colSums(t(m_trab_r))
colnames(t(m_trab_r))
rarefa <- t(m_trab_r[c("S-R-CT2","S-R-CP2","S-A-TA2","B-A-MU2"),]) #curva de rarefação
class(m_trab)
## Curva de rarefação para todos os sítios
#rarefa <- t(m_trab_r)
## Curva de rarefação para as 8 UA's com maior soma
#m_trab_r <- as.data.frame(t(m_trab_r)) #transpõe a matriz
#col_sums <- colSums(m_trab_r)
#largest_columns <- names(sort(col_sums, decreasing = TRUE)[1:8])
#rarefa <- m_trab_r[largest_columns] #curva de rarefação para as 8 UA's com maior soma

## S-A-ZA1 S-R-CC1 S-R-CT1 S-R-CP1 S-A-TA1 S-R-CT2 S-R-CP2 S-A-TA2 S-R-CT3 S-R-CP3
##      10     102     545      55     42    717     108     228    1144      68
## S-A-TA3 S-R-CT4 S-R-CP4 S-A-TA4 B-A-MU1 B-R-ET1 B-A-GU1 B-R-PC2 B-A-MU2 B-A-GU2
##      501     436     104     684    208      46      25    185     186     161
## B-R-PC3 B-A-MU3 B-A-GU3 B-R-PC4 B-A-MU4 B-A-GU4
##      371     762     535     156    1174     342
## [1] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" "S-A-TA1" "S-R-CT2" "S-R-CP2"
```

```
## [8] "S-A-TA2" "S-R-CT3" "S-R-CP3" "S-A-TA3" "S-R-CT4" "S-R-CP4" "S-A-TA4"
## [15] "B-A-MU1" "B-R-ET1" "B-A-GU1" "B-R-PC2" "B-A-MU2" "B-A-GU2" "B-R-PC3"
## [22] "B-A-MU3" "B-A-GU3" "B-R-PC4" "B-A-MU4" "B-A-GU4"
## [1] "data.frame"
```

A curva de rarefação vai ser construída para os sítios: “S-R-CT2”, “S-R-CP2”, “S-A-TA2”, “B-A-MU2”

No código abaixo o valor de `q` pode ser mudado para `q=1` para comparar a diversidade de Shannon e para `q=2` para Simpson (Figura (11.3)). A função `endpoint=` define o tamanho da amostra para o qual se deseja extrapolar a curva de rarefação. Se `endpoint=NULL`, a extração será feita para o dobro do valor amostrado.

```
library(iNEXT)
out <- iNEXT(rarefa, q = 0,
             datatype = "abundance",
             size = NULL,
             endpoint = 2000, #define o comprimento de eixo x
             knots = 40,
             se = TRUE,
             conf = 0.95,
             nboot = 50)
grafico3 <- ggiNEXT(out, type = 1, facet.var="None") +
  theme_bw() +
  labs(fill = "Áreas") +
  xlab("Número de indivíduos") +
  ylab("Riqueza de espécies") +
  theme(legend.title=element_blank()) #ver como fica com facet.var="Assemblage"
grafico3
ggsave(grafico3, dpi = 300, filename = "fig-rare1.png")

## Saving 6.5 x 4.5 in image
```

A linha sólida representa a interpolação do número de espécies observadas, e a linha tracejada mostra uma extração do que seria esperado dado um aumento no número de indivíduos coletados. A área mais clara representa o intervalo de confiança de 95%.

11.10.1 Interpretando a curva de rarefação

A curva de rarefação é uma ferramenta gráfica comumente usada em ecologia para estimar e visualizar a riqueza de espécies em uma comunidade, especialmente quando se têm dados de amostragem incompletos. Ela ajuda a responder à pergunta: “Quantas espécies diferentes podemos esperar encontrar em uma comunidade com base nas amostras que coletamos até agora?”

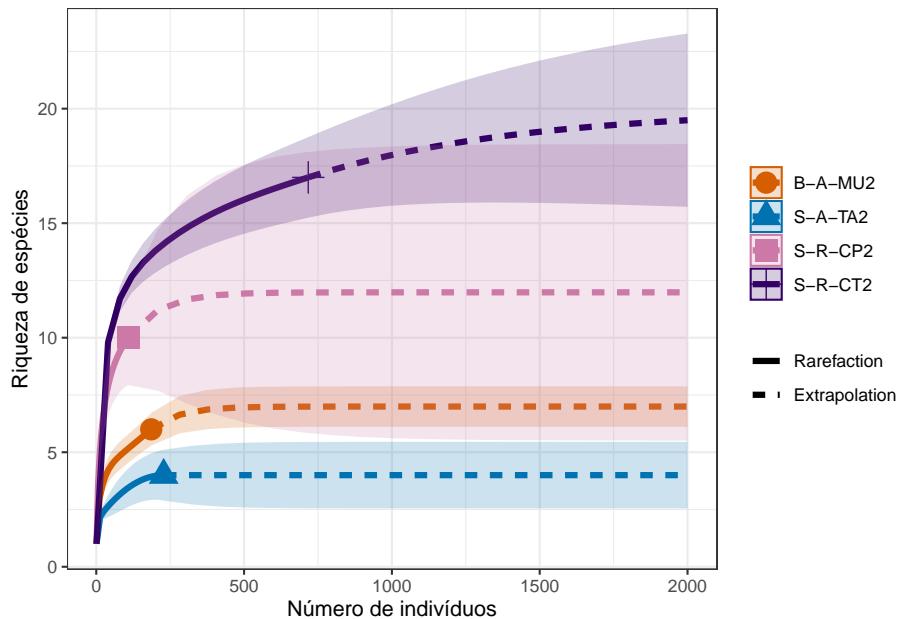


Figure 11.3: Curva de rafefação para sítios específicos. $q=0$, riqueza; $q=1$, diversidade de Shannon; $q=2$, diversidade de Simpson; $q=5$, equitabilidade.

Aqui está uma explicação passo a passo sobre como interpretar uma curva de rarefação:

1. **Eixo X (Número de Indivíduos Amostrados):**
 - O eixo horizontal (eixo X) representa o número de indivíduos ou unidades amostradas da comunidade. Isso pode ser o número de indivíduos observados, o número de amostras coletadas ou qualquer outra unidade de amostragem que seja relevante para o estudo.
2. **Eixo Y (Riqueza de Espécies):**
 - O eixo vertical (eixo Y) representa a riqueza de espécies, ou seja, o número total de espécies diferentes observadas ou estimadas nas amostras.
3. **Pontos na Curva:**
 - Cada ponto na curva de rarefação representa a riqueza de espécies estimada com base no número de indivíduos amostrados até o momento.
 - À medida que você aumenta o número de indivíduos amostrados (ou seja, move-se para a direita ao longo do eixo X), a riqueza de espécies estimada também aumenta (ou seja, move-se para cima ao longo do eixo Y).
4. **Inclinação da Curva:**
 - A inclinação da curva de rarefação é importante. Uma curva que

sobe rapidamente indica que novas espécies estão sendo encontradas à medida que mais indivíduos são amostrados. Isso sugere que a comunidade é rica em espécies, e ainda há muitas espécies não observadas.

- Uma curva que sobe lentamente sugere que a maioria das espécies já foi observada, e a riqueza de espécies está se estabilizando. Isso indica uma comunidade menos diversa ou uma amostragem mais completa.

5. Assíntota:

- A assíntota é o ponto em que a curva de rarefação começa a nivelar-se, e a adição de mais indivíduos à amostra tem um impacto mínimo na riqueza de espécies estimada. A assíntota é uma estimativa da riqueza de espécies máxima que pode ser alcançada com a amostragem disponível.

6. Interpretação:

- A interpretação da curva de rarefação depende do contexto. Se a curva ainda estiver subindo acentuadamente no ponto em que você parou de amostrar, isso sugere que a amostragem está incompleta, e mais espécies provavelmente serão encontradas com mais esforço de amostragem.
- Se a curva estiver nivelada e próxima de uma assíntota, isso sugere que a amostragem foi mais completa, e você pode ter uma estimativa confiável da riqueza de espécies na comunidade.

7. Estimativas da Riqueza:

- A curva de rarefação pode ser usada para fazer estimativas da riqueza de espécies com base nas amostras coletadas. No entanto, lembre-se de que essas são estimativas e estão sujeitas a variações amostrais. É comum calcular intervalos de confiança ao redor dessas estimativas.

Em resumo, a curva de rarefação é uma ferramenta valiosa para estimar a riqueza de espécies em uma comunidade com base em amostras coletadas. A interpretação depende da inclinação da curva, da presença de uma assíntota e do contexto do estudo. É importante lembrar que a amostragem mais completa geralmente resulta em estimativas mais confiáveis da riqueza de espécies.

11.11 Curva de acumulação de espécies

```
acumula <- specaccum(m_trab,
                      method = "random")
acumula

## Species Accumulation Curve
## Accumulation method: random, with 100 permutations
## Call: specaccum(comm = m_trab, method = "random")
##
```

```

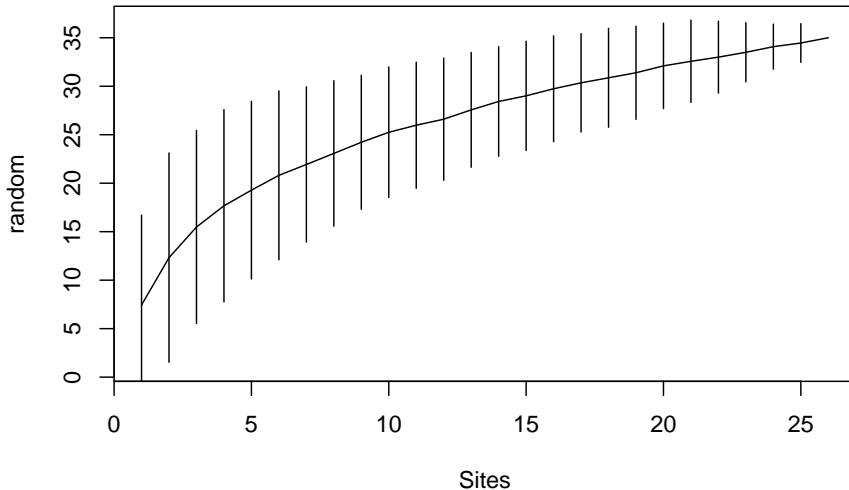
## Sites      1.0000  2.00000  3.00000  4.00000  5.00000  6.00000  7.00000  8.00000
## Richness  7.4400 12.32000 15.48000 17.67000 19.28000 20.81000 21.93000 23.07000
## sd        4.6326  5.39525  4.97972  4.95852  4.57724  4.35471  4.00317  3.75043
##
## Sites      9.00000 10.00000 11.00000 12.00000 13.00000 14.00000 15.00000 16.00000
## Richness 24.22000 25.25000 25.98000 26.60000 27.56000 28.43000 29.01000 29.74000
## sd        3.45411  3.36763  3.25353  3.14947  2.95871  2.82577  2.81947  2.7289
##
## Sites     17.0000 18.00000 19.00000 20.00000 21.00000 22.0000 23.00000 24.00000
## Richness 30.3500 30.87000 31.39000 32.10000 32.57000 33.0000 33.50000 34.08000
## sd        2.5361   2.54913  2.40326  2.20422  2.12372  1.8532   1.52753  1.16063
##
## Sites     25.00000 26
## Richness 34.46000 35
## sd        0.99919  0

```

11.11.1 Plot locais

Plotamos uma curva de acumulação de espécies simples.

```
plot(acumula)
```



Uma curva de acumulação de espécies é uma ferramenta gráfica usada para entender como a riqueza de espécies em uma comunidade aumenta à medida que mais amostras são coletadas ou observações são feitas. Ela é especialmente

útil em estudos de biodiversidade e ecologia. Aqui está uma explicação sobre como interpretar uma curva de acumulação de espécies:

1. Eixo X (Número de Amostras ou Unidades de Observação):

- O eixo horizontal (eixo X) representa o número de amostras coletadas, unidades de observação ou esforço amostral. Isso pode ser o número de áreas amostradas, pontos de observação, horas de coleta de dados ou qualquer unidade relevante para o estudo.

2. Eixo Y (Riqueza de Espécies Acumulada):

- O eixo vertical (eixo Y) representa a riqueza de espécies acumulada, ou seja, o número total de espécies diferentes observadas ou registradas até o momento.

3. Pontos na Curva:

- Cada ponto na curva de acumulação de espécies representa a riqueza de espécies acumulada com base no número de amostras coletadas ou observações feitas até aquele ponto.
- À medida que você aumenta o número de amostras (ou seja, move-se para a direita ao longo do eixo X), a riqueza de espécies acumulada também aumenta (ou seja, move-se para cima ao longo do eixo Y).

4. Inclinação da Curva:

- A inclinação da curva de acumulação de espécies é importante. Uma curva que sobe rapidamente indica que novas espécies estão sendo encontradas à medida que mais amostras são coletadas. Isso sugere que a comunidade é rica em espécies e que ainda há muitas espécies não observadas.
- Uma curva que sobe lentamente sugere que a maioria das espécies já foi observada, e a riqueza de espécies está se estabilizando. Isso indica uma comunidade menos diversa ou uma amostragem mais completa.

5. Assíntota:

- A assíntota é o ponto onde a curva de acumulação de espécies começa a nivelar-se, e a adição de mais amostras tem um impacto mínimo na riqueza de espécies acumulada. A assíntota é uma estimativa da riqueza de espécies máxima que pode ser alcançada com o esforço amostral disponível.

6. Interpretação:

- A interpretação da curva de acumulação de espécies depende do contexto. Se a curva ainda estiver subindo acentuadamente no ponto em que você parou de amostrar, isso sugere que a amostragem está incompleta, e mais espécies provavelmente serão encontradas com mais esforço amostral.
- Se a curva estiver nivelada e próxima de uma assíntota, isso sugere que a amostragem foi mais completa, e você pode ter uma estimativa confiável da riqueza de espécies na comunidade.

7. Estimativas da Riqueza:

- A curva de acumulação de espécies pode ser usada para fazer estimativas da riqueza de espécies com base nas amostras coletadas. No

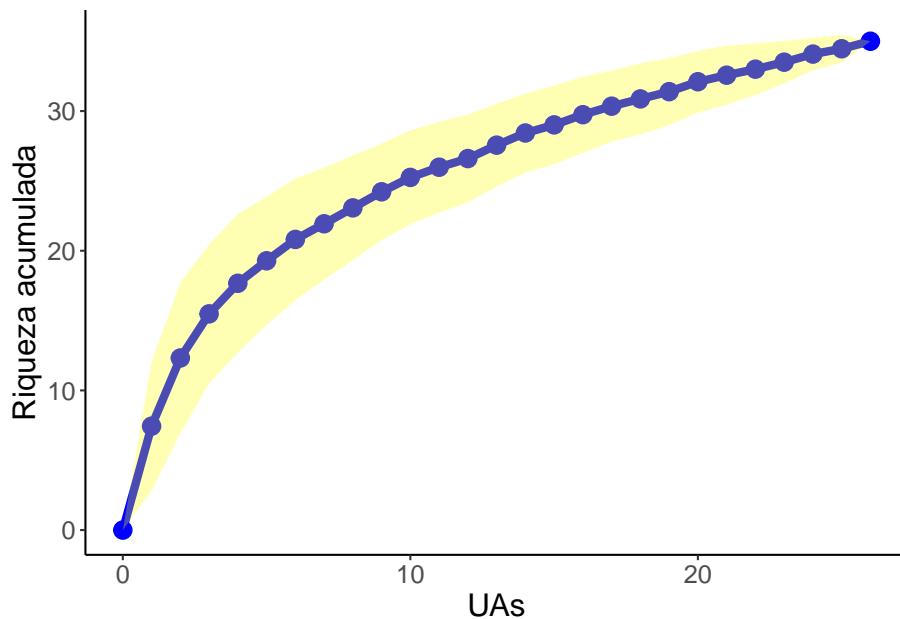
entanto, lembre-se de que essas são estimativas e estão sujeitas a variações amostrais. É comum calcular intervalos de confiança ao redor dessas estimativas.

Em resumo, a curva de acumulação de espécies é uma ferramenta valiosa para entender como a riqueza de espécies aumenta com o esforço amostral em uma comunidade. A interpretação depende da inclinação da curva, da presença de uma assíntota e do contexto do estudo. Ela ajuda a responder à pergunta: “Quantas espécies diferentes podemos esperar encontrar em uma comunidade com base nas amostras que coletamos até agora?”

Agora podemos plotar uma curva de acumulação de espécies melhorada.

```
plot_data <- data.frame("UAs" = c(0, acumula$sites),
                        "Riqueza" = c(0, acumula$richness),
                        "lower" = c(0, acumula$richness - acumula$sd),
                        "upper" = c(0, acumula$richness + acumula$sd))
gLocais <- ggplot(plot_data, aes(x = UAs, y = Riqueza)) +
  geom_point(color = "blue", size = 4) +
  geom_line(color = "blue", lwd = 2) +
  geom_ribbon(aes(ymin = lower, ymax = upper),
              linetype=2, alpha=0.3, fill = "yellow") +
  ylab("Riqueza acumulada") +
  theme_classic() +
  theme(text = element_text(size = 16))
gLocais
ggsave(gLocais, dpi = 300, filename = "fig-acum.png")
```

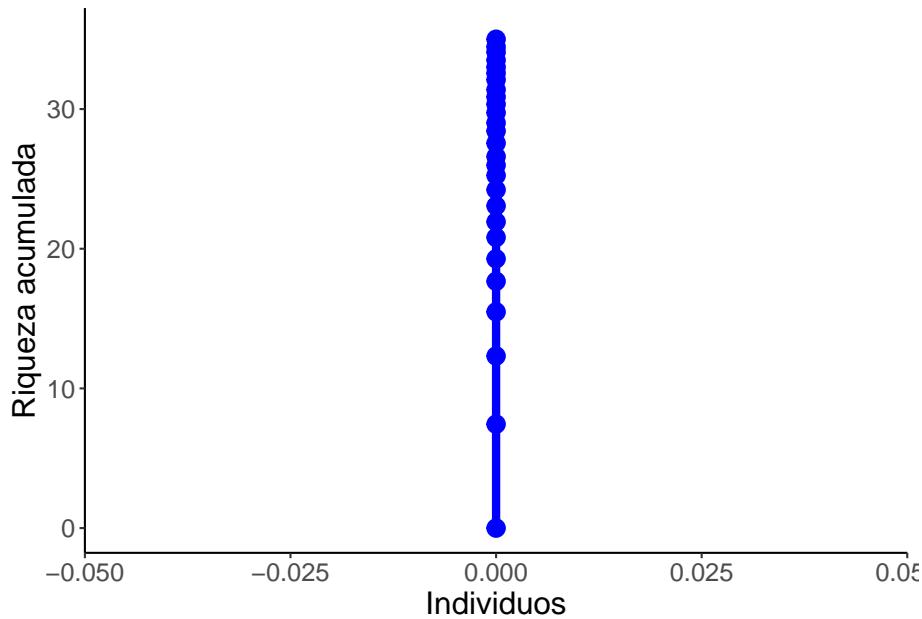
```
## Saving 6.5 x 4.5 in image
```



```

plot_data <- data.frame("Individuos" = c(0, acumula$individuals),
                       "Riqueza" = c(0, acumula$richness),
                       "lower" = c(0, acumula$richness - acumula$sd),
                       "upper" = c(0, acumula$richness + acumula$sd))
gInd <- ggplot(plot_data, aes(x = Individuos, y = Riqueza)) +
  geom_point(color = "blue", size = 4) +
  geom_line(color = "blue", lwd = 2) +
  geom_ribbon(aes(ymin = lower, ymax = upper),
              linetype=2, alpha=0.3, fill = "yellow") +
  ylab("Riqueza acumulada") +
  theme_classic() +
  theme(text = element_text(size = 16))
gInd

```



Apêndices

Criando uma tabela de espécies

Nessa etapa precisaremos da tabela de grupos. Selecionamos a aba `peixes`, porque nela constam os agrupamentos *a-priori* para os dados de peixes.

```
library(openxlsx)
t_grps <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizess/ppbio06-grupos.xlsx",
                      rowNames = T, colNames = T,
                      sheet = "peixes")
str(t_grps)
#View(ppbio)
t_grps[1:4,1:4] # [1:4,1:4] mostra apenas as linhas e colunas de 1 a 4.
```

```
## 'data.frame': 26 obs. of 4 variables:
## $ area      : chr "Serido" "Serido" "Serido" "Serido" ...
## $ ambiente   : chr "acude" "rio" "rio" "rio" ...
## $ UA        : chr "ZA" "CC" "CT" "CP" ...
## $ coleta    : num 1 1 1 1 2 2 2 3 3 ...
##           area ambiente UA coleta
## S-A-ZA1 Serido     acude ZA      1
## S-R-CC1 Serido     rio CC      1
## S-R-CT1 Serido     rio CT      1
```

```

## S-R-CP1 Serido      rio CP      1

library(dplyr)
library(tidyr)
m_trab <- m_trab %>%
  rename_with(~ gsub("-", ".", .)) #apenas troquei o hífen pelo ponto.
m <- m_trab %>%
  group_by(Area = t_grps$area,
           Habitat = t_grps$ambiente,
           Ponto = t_grps$UA) %>%
  summarise(across(where(is.numeric),
                  list(mean = mean, sd = sd)),
             .groups = 'drop') %>%
  pivot_longer(
    cols = -c(Area, Habitat, Ponto),
    names_to = c("Variable", ".value"),
    names_sep = "_"
  )
m
m <- as.data.frame(m)
m_wide <- m %>%
  mutate(stat_string = ifelse(
    Variable == "nome_da_variavel", #escolhe uma variável pra ter 3 casas decimais
    paste0(round(mean, 3), "(", round(sd, 3), ")"),
    paste0(round(mean, 1), "(", round(sd, 1), ")")
  )) %>%
  unite("Location", Area, Habitat, Ponto, sep = "_") %>%
  dplyr::select(Variable, Location, stat_string) %>% #dplyr dá conflito com MASS
  pivot_wider(names_from = Location, values_from = stat_string)

m_wide
m_wide <- as.data.frame(m_wide)
m_wide

#Exportando dados para Excel----
library(openxlsx)
write.xlsx(m_wide, file = "tabela da comunidade.xlsx", rowNames = FALSE)
wb <- loadWorkbook("tabela da comunidade.xlsx")
writeData(wb, sheet = "Sheet 1", x = m_wide)
saveWorkbook(wb, "tabela da comunidade.xlsx", overwrite = TRUE)

#Escolher sumário de uma variável----
m
var <- "as.bimac"
m[m$Variable == var, "mean"] #cada valor de var

```

```

summary(m[m$Variable == var, "mean"]) #sumário dos valores de var

## # A tibble: 315 x 6
##   Area   Habitat Ponto Variable  mean    sd
##   <chr>  <chr>    <chr>  <chr>    <dbl>  <dbl>
## 1 Buique acude GU    ap.davis  0     0
## 2 Buique acude GU    as.bimac  5.25  5.74
## 3 Buique acude GU    as.fasci  0.5   1
## 4 Buique acude GU    ch.bimac  0     0
## 5 Buique acude GU    ci.ocela  0     0
## 6 Buique acude GU    ci.orien  0     0
## 7 Buique acude GU    co.macro  0     0
## 8 Buique acude GU    co.heter  0     0
## 9 Buique acude GU    cr.menez  0     0
## 10 Buique acude GU   cu.lepid  0     0
## # i 305 more rows
## # A tibble: 35 x 10
##   Variable Buique_acude_GU Buique_acude_MU Buique_rio_ET Buique_rio_PC
##   <chr>      <chr>          <chr>          <chr>          <chr>
## 1 ap.davis  0(0)           0(0)           0(NA)          9(11.5)
## 2 as.bimac  5.2(5.7)       214.2(218.1)  3(NA)          42(34)
## 3 as.fasci  0.5(1)         0(0)           0(NA)          8(8.5)
## 4 ch.bimac  0(0)           0(0)           0(NA)          0(0)
## 5 ci.ocela  0(0)           0(0)           0(NA)          2(2)
## 6 ci.orien  0(0)           0(0)           0(NA)          0(0)
## 7 co.macro  0(0)           0(0)           0(NA)          0(0)
## 8 co.heter  0(0)           0(0)           0(NA)          0(0)
## 9 cr.menez  0(0)           0(0)           0(NA)          0(0)
## 10 cu.lepid 0(0)           0(0)           0(NA)          7(12.1)
## # i 25 more rows
## # i 5 more variables: Serido_acude_TA <chr>, Serido_acude_ZA <chr>,
## #   Serido_rio_CC <chr>, Serido_rio_CP <chr>, Serido_rio_CT <chr>
##   Variable Buique_acude_GU Buique_acude_MU Buique_rio_ET Buique_rio_PC
##   <chr>      <chr>          <chr>          <chr>          <chr>
## 1 ap.davis  0(0)           0(0)           0(NA)          9(11.5)
## 2 as.bimac  5.2(5.7)       214.2(218.1)  3(NA)          42(34)
## 3 as.fasci  0.5(1)         0(0)           0(NA)          8(8.5)
## 4 ch.bimac  0(0)           0(0)           0(NA)          0(0)
## 5 ci.ocela  0(0)           0(0)           0(NA)          2(2)
## 6 ci.orien  0(0)           0(0)           0(NA)          0(0)
## 7 co.macro  0(0)           0(0)           0(NA)          0(0)
## 8 co.heter  0(0)           0(0)           0(NA)          0(0)
## 9 cr.menez  0(0)           0(0)           0(NA)          0(0)
## 10 cu.lepid 0(0)           0(0)           0(NA)          7(12.1)
## 11 cy.gilbe 0(0)           0(0)           0(NA)          27(46.8)

```

```

## 12 ge.brasi      18(11.6)    227.8(194.3)      0(NA)     9.7(5.7)
## 13 he.margi      0(0)        0(0)        0(NA)     0(0)
## 14 ho.malab      0(0)        0.2(0.5)      0(NA)     1(1)
## 15 hy.pusar      0(0)        0(0)        0(NA)     0.3(0.6)
## 16 le.melan      0(0)        0(0)        0(NA)     0.7(1.2)
## 17 le.piau       0(0)        0(0)        0(NA)     0.3(0.6)
## 18 le.taeni      0(0)        0(0)        0(NA)     0.3(0.6)
## 19 mo.costa      0(0)        0(0)        0(NA)     0.3(0.6)
## 20 mo.lepid      0(0)        0(0)        0(NA)     0(0)
## 21 or.nilot      103.8(109.4) 4.8(4.8)      8(NA)    26.3(33.5)
## 22 pa.manag      138.2(104.4) 0(0)        1(NA)     0(0)
## 23 pimel.sp      0(0)        0(0)        0(NA)     0(0)
## 24 po.retic      0(0)        80.5(125.2) 34(NA)     0(0)
## 25 po.vivip      0(0)        54.8(75.2)   0(NA)     0(0)
## 26 pr.brevi      0(0)        0.2(0.5)     0(NA)     5(4.6)
## 27 ps.rhomb      0(0)        0(0)        0(NA)     0(0)
## 28 ps.genise     0(0)        0(0)        0(NA)     0(0)
## 29 se.heter      0(0)        0(0)        0(NA)    44.7(43.2)
## 30 se.piaba     0(0)        0(0)        0(NA)     0(0)
## 31 se.spiolo     0(0)        0(0)        0(NA)     0(0)
## 32 st.noton      0(0)        0(0)        0(NA)     0(0)
## 33 sy.marmo      0(0)        0(0)        0(NA)     0(0)
## 34 te.chalc      0(0)        0(0)        0(NA)    44.7(39.7)
## 35 tr.signa      0(0)        0(0)        0(NA)     9(12.3)

## Serido_acude_TA Serido_acude_ZA Serido_rio_CC Serido_rio_CP Serido_rio_CT
## 1      0(0)        0(NA)        0(NA)     0(0)     0(0)
## 2      174.2(174.3) 1(NA)      99(NA)    10(9)   135.5(94.5)
## 3      2(3.4)      0(NA)        0(NA)     0.2(0.5) 30.8(33.4)
## 4      175.5(115.2) 0(NA)        0(NA)     0(0)     0.8(1.5)
## 5      0(0)        0(NA)        0(NA)    16(17)   0(0)
## 6      0(0)        0(NA)        0(NA)    9.8(10.2) 26(30.2)
## 7      0.5(1)      0(NA)        0(NA)     0(0)     0(0)
## 8      0(0)        0(NA)        0(NA)     0(0)     0.2(0.5)
## 9      0.2(0.5)    0(NA)        0(NA)     0(0)     6.8(5.6)
## 10     0(0)        0(NA)        0(NA)     0(0)     0(0)
## 11     0(0)        0(NA)        0(NA)     0(0)    12.5(25)
## 12     0.2(0.5)    0(NA)        0(NA)     0(0)     1.8(1.5)
## 13     0(0)        0(NA)        0(NA)     0(0)     0.5(0.6)
## 14     7.8(9)      0(NA)        0(NA)     5.2(3.4) 13.2(13.7)
## 15     0(0)        0(NA)        0(NA)    1(1.2)   16.5(18)
## 16     0(0)        0(NA)        0(NA)     0(0)     0(0)
## 17     0.5(1)      0(NA)        0(NA)    1.5(1.3) 1.5(1.3)
## 18     0(0)        0(NA)        0(NA)     0(0)     0(0)
## 19     0(0)        0(NA)        0(NA)     0(0)     0(0)
## 20     0(0)        0(NA)        1(NA)     0(0)    10(19.3)
## 21     0.2(0.5)    0(NA)        2(NA)     0(0)    81(42.2)

```

```

## 22      0(0)      0(NA)      0(NA)      0(0)      0(0)
## 23      0(0)      0(NA)      0(NA)      0(0)      1.5(3)
## 24      0(0)      0(NA)      0(NA)      0(0)      6.2(9.5)
## 25      0(0)      0(NA)      0(NA)      0(NA)    34.2(31.9) 155.5(143)
## 26      1.5(1.3)  9(NA)      0(NA)      1.2(2.5) 60.8(72.7)
## 27      0(0)      0(NA)      0(NA)      0(0)      0.2(0.5)
## 28      0(0)      0(NA)      0(NA)      0(0)      0.2(0.5)
## 29      1(2)      0(NA)      0(NA)      4.2(6.7) 35.2(23.7)
## 30      0(0)      0(NA)      0(NA)      0(0)      17(34)
## 31      0(0)      0(NA)      0(NA)      0(0)      0.2(0.5)
## 32      0(0)      0(NA)      0(NA)      0(0)      51.2(49.8)
## 33      0(0)      0(NA)      0(NA)      0.2(0.5) 0(0)
## 34      0(0)      0(NA)      0(NA)      0(0)      0(0)
## 35      0(0)      0(NA)      0(NA)      0(0)      45.2(64)

##          Area Habitat Ponto Variable   mean     sd
## 1       Buique acude   GU ap.davis 0.0000000 0.0000000
## 2       Buique acude   GU as.bimac 5.2500000 5.7373048
## 3       Buique acude   GU as.fasci 0.5000000 1.0000000
## 4       Buique acude   GU ch.bimac 0.0000000 0.0000000
## 5       Buique acude   GU ci.ocela 0.0000000 0.0000000
## 6       Buique acude   GU ci.orien 0.0000000 0.0000000
## 7       Buique acude   GU co.macro 0.0000000 0.0000000
## 8       Buique acude   GU co.heter 0.0000000 0.0000000
## 9       Buique acude   GU cr.menez 0.0000000 0.0000000
## 10      Buique acude   GU cu.lepid 0.0000000 0.0000000
## 11      Buique acude   GU cy.gilbe 0.0000000 0.0000000
## 12      Buique acude   GU ge.brasi 18.0000000 11.6332856
## 13      Buique acude   GU he.margi 0.0000000 0.0000000
## 14      Buique acude   GU ho.malab 0.0000000 0.0000000
## 15      Buique acude   GU hy.pusar 0.0000000 0.0000000
## 16      Buique acude   GU le.melan 0.0000000 0.0000000
## 17      Buique acude   GU le.piau 0.0000000 0.0000000
## 18      Buique acude   GU le.taeni 0.0000000 0.0000000
## 19      Buique acude   GU mo.costा 0.0000000 0.0000000
## 20      Buique acude   GU mo.lepid 0.0000000 0.0000000
## 21      Buique acude   GU or.nilot 103.7500000 109.3903561
## 22      Buique acude   GU pa.manag 138.2500000 104.3595547
## 23      Buique acude   GU pimel.sp 0.0000000 0.0000000
## 24      Buique acude   GU po.retic 0.0000000 0.0000000
## 25      Buique acude   GU po.vivip 0.0000000 0.0000000
## 26      Buique acude   GU pr.brevi 0.0000000 0.0000000
## 27      Buique acude   GU ps.rhomb 0.0000000 0.0000000
## 28      Buique acude   GU ps.genise 0.0000000 0.0000000
## 29      Buique acude   GU se.heter 0.0000000 0.0000000
## 30      Buique acude   GU se.piaba 0.0000000 0.0000000
## 31      Buique acude   GU se.spiло 0.0000000 0.0000000

```

```

## 32 Buique acude GU st.noton 0.0000000 0.0000000
## 33 Buique acude GU sy.marmo 0.0000000 0.0000000
## 34 Buique acude GU te.chalc 0.0000000 0.0000000
## 35 Buique acude GU tr.signa 0.0000000 0.0000000
## 36 Buique acude MU ap.davis 0.0000000 0.0000000
## 37 Buique acude MU as.bimac 214.2500000 218.0815979
## 38 Buique acude MU as.fasci 0.0000000 0.0000000
## 39 Buique acude MU ch.bimac 0.0000000 0.0000000
## 40 Buique acude MU ci.ocela 0.0000000 0.0000000
## 41 Buique acude MU ci.orien 0.0000000 0.0000000
## 42 Buique acude MU co.macro 0.0000000 0.0000000
## 43 Buique acude MU co.heter 0.0000000 0.0000000
## 44 Buique acude MU cr.menez 0.0000000 0.0000000
## 45 Buique acude MU cu.lepid 0.0000000 0.0000000
## 46 Buique acude MU cy.gilbe 0.0000000 0.0000000
## 47 Buique acude MU ge.brasi 227.7500000 194.2633522
## 48 Buique acude MU he.margi 0.0000000 0.0000000
## 49 Buique acude MU ho.malab 0.2500000 0.5000000
## 50 Buique acude MU hy.pusar 0.0000000 0.0000000
## 51 Buique acude MU le.melan 0.0000000 0.0000000
## 52 Buique acude MU le.piau 0.0000000 0.0000000
## 53 Buique acude MU le.taeni 0.0000000 0.0000000
## 54 Buique acude MU mo.costa 0.0000000 0.0000000
## 55 Buique acude MU mo.lepid 0.0000000 0.0000000
## 56 Buique acude MU or.nilot 4.7500000 4.7871355
## 57 Buique acude MU pa.manag 0.0000000 0.0000000
## 58 Buique acude MU pimel.sp 0.0000000 0.0000000
## 59 Buique acude MU po.retic 80.5000000 125.2344468
## 60 Buique acude MU po.vivip 54.7500000 75.1592753
## 61 Buique acude MU pr.brevi 0.2500000 0.5000000
## 62 Buique acude MU ps.rhomb 0.0000000 0.0000000
## 63 Buique acude MU ps.genise 0.0000000 0.0000000
## 64 Buique acude MU se.heter 0.0000000 0.0000000
## 65 Buique acude MU se.piaba 0.0000000 0.0000000
## 66 Buique acude MU se.spilo 0.0000000 0.0000000
## 67 Buique acude MU st.noton 0.0000000 0.0000000
## 68 Buique acude MU sy.marmo 0.0000000 0.0000000
## 69 Buique acude MU te.chalc 0.0000000 0.0000000
## 70 Buique acude MU tr.signa 0.0000000 0.0000000
## 71 Buique rio ET ap.davis 0.0000000 NA
## 72 Buique rio ET as.bimac 3.0000000 NA
## 73 Buique rio ET as.fasci 0.0000000 NA
## 74 Buique rio ET ch.bimac 0.0000000 NA
## 75 Buique rio ET ci.ocela 0.0000000 NA
## 76 Buique rio ET ci.orien 0.0000000 NA
## 77 Buique rio ET co.macro 0.0000000 NA

```

```

## 78 Buique    rio   ET  co.heter  0.0000000      NA
## 79 Buique    rio   ET  cr.menez  0.0000000      NA
## 80 Buique    rio   ET  cu.lepid  0.0000000      NA
## 81 Buique    rio   ET  cy.gilbe  0.0000000      NA
## 82 Buique    rio   ET  ge.brasi  0.0000000      NA
## 83 Buique    rio   ET  he.margi  0.0000000      NA
## 84 Buique    rio   ET  ho.malab  0.0000000      NA
## 85 Buique    rio   ET  hy.pusar  0.0000000      NA
## 86 Buique    rio   ET  le.melan  0.0000000      NA
## 87 Buique    rio   ET  le.piau   0.0000000      NA
## 88 Buique    rio   ET  le.taeni  0.0000000      NA
## 89 Buique    rio   ET  mo.costa  0.0000000      NA
## 90 Buique    rio   ET  mo.lepid  0.0000000      NA
## 91 Buique    rio   ET  or.nilot  8.0000000      NA
## 92 Buique    rio   ET  pa.manag  1.0000000      NA
## 93 Buique    rio   ET  pimel.sp  0.0000000      NA
## 94 Buique    rio   ET  po.retic  34.0000000     NA
## 95 Buique    rio   ET  po.vivip  0.0000000      NA
## 96 Buique    rio   ET  pr.brevi  0.0000000      NA
## 97 Buique    rio   ET  ps.rhomb  0.0000000      NA
## 98 Buique    rio   ET  ps.genise 0.0000000      NA
## 99 Buique    rio   ET  se.heter  0.0000000      NA
## 100 Buique   rio   ET  se.piaba 0.0000000      NA
## 101 Buique   rio   ET  se.spilo  0.0000000      NA
## 102 Buique   rio   ET  st.noton  0.0000000      NA
## 103 Buique   rio   ET  sy.marmo 0.0000000      NA
## 104 Buique   rio   ET  te.chalc  0.0000000      NA
## 105 Buique   rio   ET  tr.signa  0.0000000      NA
## 106 Buique   rio   PC  ap.davis  9.0000000  11.5325626
## 107 Buique   rio   PC  as.bimac  42.0000000 34.0440891
## 108 Buique   rio   PC  as.fasci  8.0000000  8.5440037
## 109 Buique   rio   PC  ch.bimac  0.0000000  0.0000000
## 110 Buique   rio   PC  ci.ocela  2.0000000  2.0000000
## 111 Buique   rio   PC  ci.orien  0.0000000  0.0000000
## 112 Buique   rio   PC  co.macro  0.0000000  0.0000000
## 113 Buique   rio   PC  co.heter  0.0000000  0.0000000
## 114 Buique   rio   PC  cr.menez  0.0000000  0.0000000
## 115 Buique   rio   PC  cu.lepid  7.0000000 12.1243557
## 116 Buique   rio   PC  cy.gilbe  27.0000000 46.7653718
## 117 Buique   rio   PC  ge.brasi  9.6666667  5.6862407
## 118 Buique   rio   PC  he.margi  0.0000000  0.0000000
## 119 Buique   rio   PC  ho.malab  1.0000000  1.0000000
## 120 Buique   rio   PC  hy.pusar  0.3333333  0.5773503
## 121 Buique   rio   PC  le.melan  0.6666667  1.1547005
## 122 Buique   rio   PC  le.piau   0.3333333  0.5773503
## 123 Buique   rio   PC  le.taeni  0.3333333  0.5773503

```

```

## 124 Buique    rio    PC   mo.costa  0.3333333  0.5773503
## 125 Buique    rio    PC   mo.lepid  0.0000000  0.0000000
## 126 Buique    rio    PC   or.nilot  26.3333333 33.5459883
## 127 Buique    rio    PC   pa.manag  0.0000000  0.0000000
## 128 Buique    rio    PC   pimel.sp  0.0000000  0.0000000
## 129 Buique    rio    PC   po.retic  0.0000000  0.0000000
## 130 Buique    rio    PC   po.vivip  0.0000000  0.0000000
## 131 Buique    rio    PC   pr.brevi  5.0000000  4.5825757
## 132 Buique    rio    PC   ps.rhomb  0.0000000  0.0000000
## 133 Buique    rio    PC   ps.genise 0.0000000  0.0000000
## 134 Buique    rio    PC   se.heter  44.6666667 43.1547603
## 135 Buique    rio    PC   se.piaba  0.0000000  0.0000000
## 136 Buique    rio    PC   se.spilo  0.0000000  0.0000000
## 137 Buique    rio    PC   st.noton  0.0000000  0.0000000
## 138 Buique    rio    PC   sy.marmo  0.0000000  0.0000000
## 139 Buique    rio    PC   te.chalc  44.6666667 39.7156560
## 140 Buique    rio    PC   tr.signa  9.0000000 12.2882057
## 141 Serido    acude TA   ap.davis  0.0000000  0.0000000
## 142 Serido    acude TA   as.bimac 174.2500000 174.3413797
## 143 Serido    acude TA   as.fasci  2.0000000  3.3665016
## 144 Serido    acude TA   ch.bimac 175.5000000 115.2171862
## 145 Serido    acude TA   ci.ocela  0.0000000  0.0000000
## 146 Serido    acude TA   ci.orien  0.0000000  0.0000000
## 147 Serido    acude TA   co.macro  0.5000000  1.0000000
## 148 Serido    acude TA   co.heter  0.0000000  0.0000000
## 149 Serido    acude TA   cr.menez 0.2500000  0.5000000
## 150 Serido    acude TA   cu.lepid  0.0000000  0.0000000
## 151 Serido    acude TA   cy.gilbe  0.0000000  0.0000000
## 152 Serido    acude TA   ge.brasi 0.2500000  0.5000000
## 153 Serido    acude TA   he.margi 0.0000000  0.0000000
## 154 Serido    acude TA   ho.malab 7.7500000  9.0323493
## 155 Serido    acude TA   hy.pusar 0.0000000  0.0000000
## 156 Serido    acude TA   le.melan 0.0000000  0.0000000
## 157 Serido    acude TA   le.piau  0.5000000  1.0000000
## 158 Serido    acude TA   le.taeni 0.0000000  0.0000000
## 159 Serido    acude TA   mo.costa 0.0000000  0.0000000
## 160 Serido    acude TA   mo.lepid 0.0000000  0.0000000
## 161 Serido    acude TA   or.nilot 0.2500000  0.5000000
## 162 Serido    acude TA   pa.manag 0.0000000  0.0000000
## 163 Serido    acude TA   pimel.sp 0.0000000  0.0000000
## 164 Serido    acude TA   po.retic 0.0000000  0.0000000
## 165 Serido    acude TA   po.vivip 0.0000000  0.0000000
## 166 Serido    acude TA   pr.brevi 1.5000000  1.2909944
## 167 Serido    acude TA   ps.rhomb 0.0000000  0.0000000
## 168 Serido    acude TA   ps.genise 0.0000000  0.0000000
## 169 Serido    acude TA   se.heter 1.0000000  2.0000000

```

```

## 170 Serido acude TA se.piaba 0.0000000 0.0000000
## 171 Serido acude TA se.spilo 0.0000000 0.0000000
## 172 Serido acude TA st.noton 0.0000000 0.0000000
## 173 Serido acude TA sy.marmo 0.0000000 0.0000000
## 174 Serido acude TA te.chalc 0.0000000 0.0000000
## 175 Serido acude TA tr.signa 0.0000000 0.0000000
## 176 Serido acude ZA ap.davis 0.0000000 NA
## 177 Serido acude ZA as.bimac 1.0000000 NA
## 178 Serido acude ZA as.fasci 0.0000000 NA
## 179 Serido acude ZA ch.bimac 0.0000000 NA
## 180 Serido acude ZA ci.ocela 0.0000000 NA
## 181 Serido acude ZA ci.orien 0.0000000 NA
## 182 Serido acude ZA co.macro 0.0000000 NA
## 183 Serido acude ZA co.heter 0.0000000 NA
## 184 Serido acude ZA cr.menez 0.0000000 NA
## 185 Serido acude ZA cu.lepid 0.0000000 NA
## 186 Serido acude ZA cy.gilbe 0.0000000 NA
## 187 Serido acude ZA ge.brasi 0.0000000 NA
## 188 Serido acude ZA he.margi 0.0000000 NA
## 189 Serido acude ZA ho.malab 0.0000000 NA
## 190 Serido acude ZA hy.pusar 0.0000000 NA
## 191 Serido acude ZA le.melan 0.0000000 NA
## 192 Serido acude ZA le.piau 0.0000000 NA
## 193 Serido acude ZA le.taeni 0.0000000 NA
## 194 Serido acude ZA mo.costa 0.0000000 NA
## 195 Serido acude ZA mo.lepid 0.0000000 NA
## 196 Serido acude ZA or.nilot 0.0000000 NA
## 197 Serido acude ZA pa.manag 0.0000000 NA
## 198 Serido acude ZA pimel.sp 0.0000000 NA
## 199 Serido acude ZA po.retic 0.0000000 NA
## 200 Serido acude ZA po.vivip 0.0000000 NA
## 201 Serido acude ZA pr.brevi 9.0000000 NA
## 202 Serido acude ZA ps.rhomb 0.0000000 NA
## 203 Serido acude ZA ps.genise 0.0000000 NA
## 204 Serido acude ZA se.heter 0.0000000 NA
## 205 Serido acude ZA se.piaba 0.0000000 NA
## 206 Serido acude ZA se.spilo 0.0000000 NA
## 207 Serido acude ZA st.noton 0.0000000 NA
## 208 Serido acude ZA sy.marmo 0.0000000 NA
## 209 Serido acude ZA te.chalc 0.0000000 NA
## 210 Serido acude ZA tr.signa 0.0000000 NA
## 211 Serido rio CC ap.davis 0.0000000 NA
## 212 Serido rio CC as.bimac 99.0000000 NA
## 213 Serido rio CC as.fasci 0.0000000 NA
## 214 Serido rio CC ch.bimac 0.0000000 NA
## 215 Serido rio CC ci.ocela 0.0000000 NA

```

```

## 216 Serido    rio   CC  ci.orien  0.0000000  NA
## 217 Serido    rio   CC  co.macro  0.0000000  NA
## 218 Serido    rio   CC  co.heter  0.0000000  NA
## 219 Serido    rio   CC  cr.menez  0.0000000  NA
## 220 Serido    rio   CC  cu.lepid  0.0000000  NA
## 221 Serido    rio   CC  cy.gilbe  0.0000000  NA
## 222 Serido    rio   CC  ge.brasi  0.0000000  NA
## 223 Serido    rio   CC  he.margi  0.0000000  NA
## 224 Serido    rio   CC  ho.malab  0.0000000  NA
## 225 Serido    rio   CC  hy.pusar  0.0000000  NA
## 226 Serido    rio   CC  le.melan  0.0000000  NA
## 227 Serido    rio   CC  le.piau  0.0000000  NA
## 228 Serido    rio   CC  le.taeni  0.0000000  NA
## 229 Serido    rio   CC  mo.costा  0.0000000  NA
## 230 Serido    rio   CC  mo.lepid  1.0000000  NA
## 231 Serido    rio   CC  or.nilot  2.0000000  NA
## 232 Serido    rio   CC  pa.manag  0.0000000  NA
## 233 Serido    rio   CC  pimel.sp  0.0000000  NA
## 234 Serido    rio   CC  po.retic  0.0000000  NA
## 235 Serido    rio   CC  po.vivip  0.0000000  NA
## 236 Serido    rio   CC  pr.brevi  0.0000000  NA
## 237 Serido    rio   CC  ps.rhomb  0.0000000  NA
## 238 Serido    rio   CC  ps.genise 0.0000000  NA
## 239 Serido    rio   CC  se.heter  0.0000000  NA
## 240 Serido    rio   CC  se.piaba  0.0000000  NA
## 241 Serido    rio   CC  se.spilo  0.0000000  NA
## 242 Serido    rio   CC  st.noton  0.0000000  NA
## 243 Serido    rio   CC  sy.marmo  0.0000000  NA
## 244 Serido    rio   CC  te.chalc  0.0000000  NA
## 245 Serido    rio   CC  tr.signa  0.0000000  NA
## 246 Serido    rio   CP  ap.davis  0.0000000  0.0000000
## 247 Serido    rio   CP  as.bimac  10.0000000  8.9814624
## 248 Serido    rio   CP  as.fasci  0.2500000  0.5000000
## 249 Serido    rio   CP  ch.bimac  0.0000000  0.0000000
## 250 Serido    rio   CP  ci.ocela  16.0000000  16.9901932
## 251 Serido    rio   CP  ci.orien  9.7500000  10.2102889
## 252 Serido    rio   CP  co.macro  0.0000000  0.0000000
## 253 Serido    rio   CP  co.heter  0.0000000  0.0000000
## 254 Serido    rio   CP  cr.menez  0.0000000  0.0000000
## 255 Serido    rio   CP  cu.lepid  0.0000000  0.0000000
## 256 Serido    rio   CP  cy.gilbe  0.0000000  0.0000000
## 257 Serido    rio   CP  ge.brasi  0.0000000  0.0000000
## 258 Serido    rio   CP  he.margi  0.0000000  0.0000000
## 259 Serido    rio   CP  ho.malab  5.2500000  3.4034296
## 260 Serido    rio   CP  hy.pusar  1.0000000  1.1547005
## 261 Serido    rio   CP  le.melan  0.0000000  0.0000000

```

```

## 262 Serido    rio    CP    le.piau   1.5000000  1.2909944
## 263 Serido    rio    CP    le.taeni  0.0000000  0.0000000
## 264 Serido    rio    CP    mo.costa  0.0000000  0.0000000
## 265 Serido    rio    CP    mo.lepid  0.0000000  0.0000000
## 266 Serido    rio    CP    or.nilot  0.0000000  0.0000000
## 267 Serido    rio    CP    pa.manag  0.0000000  0.0000000
## 268 Serido    rio    CP    pimel.sp  0.0000000  0.0000000
## 269 Serido    rio    CP    po.retic  0.0000000  0.0000000
## 270 Serido    rio    CP    po.vivip  34.2500000 31.9204741
## 271 Serido    rio    CP    pr.brevi  1.2500000  2.5000000
## 272 Serido    rio    CP    ps.rhomb  0.0000000  0.0000000
## 273 Serido    rio    CP    ps.genise 0.0000000  0.0000000
## 274 Serido    rio    CP    se.heter  4.2500000  6.6520673
## 275 Serido    rio    CP    se.piaba  0.0000000  0.0000000
## 276 Serido    rio    CP    se.spilo  0.0000000  0.0000000
## 277 Serido    rio    CP    st.noton  0.0000000  0.0000000
## 278 Serido    rio    CP    sy.marmo  0.2500000  0.5000000
## 279 Serido    rio    CP    te.chalc  0.0000000  0.0000000
## 280 Serido    rio    CP    tr.signa  0.0000000  0.0000000
## 281 Serido    rio    CT    ap.davis  0.0000000  0.0000000
## 282 Serido    rio    CT    as.bimac 135.5000000 94.5074953
## 283 Serido    rio    CT    as.fasci  30.7500000 33.4103277
## 284 Serido    rio    CT    ch.bimac  0.7500000  1.5000000
## 285 Serido    rio    CT    ci.ocela  0.0000000  0.0000000
## 286 Serido    rio    CT    ci.orien  26.0000000 30.1772541
## 287 Serido    rio    CT    co.macro  0.0000000  0.0000000
## 288 Serido    rio    CT    co.heter  0.2500000  0.5000000
## 289 Serido    rio    CT    cr.menez 6.7500000  5.6199051
## 290 Serido    rio    CT    cu.lepid  0.0000000  0.0000000
## 291 Serido    rio    CT    cy.gilbe  12.5000000 25.0000000
## 292 Serido    rio    CT    ge.brasi  1.7500000  1.5000000
## 293 Serido    rio    CT    he.margi  0.5000000  0.5773503
## 294 Serido    rio    CT    ho.malab  13.2500000 13.7204227
## 295 Serido    rio    CT    hy.pusar  16.5000000 17.9907384
## 296 Serido    rio    CT    le.melan  0.0000000  0.0000000
## 297 Serido    rio    CT    le.piau  1.5000000  1.2909944
## 298 Serido    rio    CT    le.taeni  0.0000000  0.0000000
## 299 Serido    rio    CT    mo.costa  0.0000000  0.0000000
## 300 Serido    rio    CT    mo.lepid 10.0000000 19.3390796
## 301 Serido    rio    CT    or.nilot  81.0000000 42.2453153
## 302 Serido    rio    CT    pa.manag  0.0000000  0.0000000
## 303 Serido    rio    CT    pimel.sp  1.5000000  3.0000000
## 304 Serido    rio    CT    po.retic  6.2500000  9.4648472
## 305 Serido    rio    CT    po.vivip 155.5000000 143.0489427
## 306 Serido    rio    CT    pr.brevi  60.7500000 72.7203548
## 307 Serido    rio    CT    ps.rhomb  0.2500000  0.5000000

```

```

## 308 Serido    rio    CT ps.genise   0.2500000  0.5000000
## 309 Serido    rio    CT se.heter   35.2500000 23.6836793
## 310 Serido    rio    CT se.piaba  17.0000000 34.0000000
## 311 Serido    rio    CT se.spilo   0.2500000  0.5000000
## 312 Serido    rio    CT st.noton  51.2500000 49.8021084
## 313 Serido    rio    CT sy.marmo  0.0000000  0.0000000
## 314 Serido    rio    CT te.chalc  0.0000000  0.0000000
## 315 Serido    rio    CT tr.signa 45.2500000 64.0019531
## [1] 5.25 214.25 3.00 42.00 174.25 1.00 99.00 10.00 135.50
##      Min. 1st Qu. Median Mean 3rd Qu. Max.
##      1.00 5.25 42.00 76.03 135.50 214.25

```

Descritores da estrutura da comunidade: UA's

Sum, soma; mean, média; DP, desvio padrão da média; Max, maior valor; Min, menor valor; S, riqueza (ou frequência de ocorrência na matriz transposta); E, índice de equitabilidade de Pielou (PIELOU, 1975); H, índice de diversidade de Shannon (LUDWIG; REYNOLDS, 1988); D, índice de diversidade de Simpson (HURLBERT, 1971).

```

#Matriz transposta
m_trab <- t(m_trab) #transpõe a matriz de volta para comunitaria
Sum <- rowSums(m_trab)
Sum <- apply(m_trab,1,sum)
Mean <- rowMeans(m_trab)
Mean <- apply(m_trab,1,mean)
DP <- apply(m_trab,1,sd)
Max <- apply(m_trab,1,max)
Min <- apply(m_trab,1,min)
#library(vegan)
bin <- decostand(m_trab,"pa")
S <- apply(bin,1,sum)
Riqueza <- specnumber(m_trab)
Riqueza_total <- specnumber(colSums(m_trab))
H <- diversity(m_trab)
D <- diversity(m_trab, "simpson")
D[is.na(D)] <- 0 #substitui NA ou NaN por 0
E <- H/log(specnumber(m_trab))
E[is.na(E)] <- 0 #substitui NA ou NaN por 0
#library(moments)
Assimetria <- apply(m_trab,1,skewness)
Curtose <- apply(m_trab,1,kurtosis)
m_trab <- t(m_trab) #traz de volta
Descritores1 <- cbind(Sum, Mean, DP, Max, Min, S, E, H, D)
Descritores1 <- as.data.frame(Descritores1)

```

```

Descritores1
#Descritores1 <- Descritores1 %>% rownames_to_column(var="UA`s") #da nome a primeira coluna
SomaTotalD <- apply(Descritores1,2,sum)
SomaTotalD
MediaTotalD <- apply(Descritores1,2,mean)
MediaTotalD
DPTotalD <- apply(Descritores1,2,sd)
DPTotalD
Descritores2 <- cbind(SomaTotalD, MediaTotalD, DPTotalD)
Descritores2 <- as.data.frame(Descritores2)
Descritores2 <- t(Descritores2)
Descritores2
DescritoresFinal <- rbind(Descritores1, Descritores2)
DescritoresFinal
DescritoresFinal <- round (DescritoresFinal, 2)
#Fazendo uma tabela
#library(gt)
df <- DescritoresFinal
ncol(df); nrow(df) #no. de N colunas x M linhas
df <- cbind(UA = rownames(df), df)
gt(df, caption = "Descritores da diversidade por Unidade Amostral (UA). Sum, soma; mean, media; sd, desvio padrao")

```

Descritores da normalidade

```

Normalidade1 <- cbind(Assimetria, Curtose)
Normalidade1 <- as.data.frame(Normalidade1)
Normalidade1 <- na.omit(Normalidade1) #remove NA e NaN
Normalidade1
SomaTotalN <- apply(Normalidade1,2,sum)
SomaTotalN
MediaTotalN <- apply(Normalidade1,2,mean)
MediaTotalN
DPTotalN <- apply(Normalidade1,2,sd)
DPTotalN
Normalidade2<-cbind(SomaTotalN, MediaTotalN, DPTotalN)
Normalidade2<-as.data.frame(Normalidade2)
Normalidade2 <- t(Normalidade2) #"t" transpõe a matriz
Normalidade2
NormalidadeFinal <- rbind(Normalidade1, Normalidade2)
NormalidadeFinal
NormalidadeFinal <- round(NormalidadeFinal, 2)
#Fazendo uma tabela
nf <- NormalidadeFinal

```

```
ncol(nf); nrow(nf) #no. de N colunas x M linhas
nf <- cbind(UA = rownames(nf), nf)
gt(nf, caption = "Descritores da normalidade por Unidade Amostral (UA)")
```

A função `fix(nome da matriz)` dá acesso ao grid da matriz criada para manipulação dos dados numéricos.

Salvando as tabelas criadas em txt direto no diretório de trabalho

```
write.table(data.frame("Spp"=rownames(DescritoresFinal),
                      DescritoresFinal),
            "DescritoresUA.csv",
            row.names=FALSE,
            sep="\t")
write.table(data.frame("Spp"=rownames(NormalidadeFinal),
                      NormalidadeFinal),
            "NormalidadeUA.txt",
            row.names=FALSE,
            sep="\t")
```

Sites consultados

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executa-lo. Lembre de remover os `#` ou `##` caso necessite executar essas linhas.

```
## dev.off() #apaga os graficos, se houver algum
## rm(list=ls(all=TRUE)) #limpa a memória
## cat("\014") #limpa o console
## install.packages("vegan")
## install.packages("moments")
## install.packages("ggplot2")
## install.packages("dplyr")
## install.packages("tidyverse")
## install.packages("tibble")
## install.packages("tidyverse") #atente para alguma msg de erro qdo executar essa linha
## install.packages("forcats")
## install.packages("iNEXT")
## install.packages("openxlsx")
```

```

## install.packages("gt")
## library(tibble); library(tidyverse); library(forcats); library(openxlsx); library(R
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
#dir <- getwd() #criamos um vetor com o diretório de trabalho
#shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06.xlsx"
                  rowNames = T, colNames = T,
                  sheet = "Sheet1")
str(ppbio)
#View(ppbio)
ppbio[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
## #View(ppbio)
## print(ppbio[1:8,1:8])
## ppbio[1:10,1:10]
## str(ppbio)
## mode(ppbio)
## class(ppbio)
m_trab <- ppbio
m_trab <- t(ppbio)
str(m_trab)
#View(m_trab)
m_trab
print(m_trab[1:5,1:5])
m_trab[1:5,1:5]
str(m_trab)
mode(m_trab)
class(m_trab)
range(m_trab) #menor e maior valores
length(m_trab) #no. de colunas
ncol(m_trab) #no. de N colunas
nrow(m_trab) #no. de M linhas
sum(lengths(m_trab)) #soma os nos. de colunas
length(as.matrix(m_trab)) #tamanho da matriz m x n
sum(m_trab == 0) #número de observações igual a zero
sum(m_trab > 0) #número de observações maiores que zero
#calculando a proporção de zeros na matriz
zeros <- (sum(m_trab == 0)/length(as.matrix(m_trab)))*100
zeros
tamanho <- data.frame(
  Função = c("range", "lenght", "n cols", "m linhas", "Tamanho", "Tamanho",
            "Zeros", "Nao zeros", "% Zeros"),
  Resultado = c(paste(range(m_trab), collapse = " - "),
               length(m_trab), ncol(m_trab),

```

```

nrow(m_trab), sum(lengths(m_trab)),
length(as.matrix(m_trab)), sum(m_trab == 0),
sum(m_trab > 0), round(zeros, 1)))
tamanho
knitr::kable(tamanho, format = "markdown", caption = "Resumo das informações sobre o tamanho da base de dados")
#?apply
Sum <- rowSums(m_trab)
#ou
Sum <- apply(m_trab, 1, sum)
Sum
## Media
Mean <- rowMeans(m_trab)
Mean
## Ou
Mean <- apply(m_trab, 1, mean)
Mean
## Desvio padrão
DP <- apply(m_trab, 1, sd)
DP
## Máximo
Max <- apply(m_trab, 1, max)
Max
## Mínimo
Min <- apply(m_trab, 1, min)
Min
library(vegan)
bin <- decostand(m_trab, "pa")
bin[1:10, 1:10]
S <- apply(bin, 1, sum)
S
#OU
Riqueza <- specnumber(m_trab)
Riqueza
Riqueza_total <- specnumber(colSums(m_trab))
Riqueza_total
H <- diversity(m_trab, index = "shannon")
H
D <- diversity(m_trab, "simpson")
D
D[is.na(D)] <- 0 #substitui NA ou NaN por 0
D
E <- H / log(specnumber(m_trab))
E
E[is.na(E)] <- 0 #substitui NA ou NaN por 0

```

```

E
library(moments)
Assimetria <- apply(m_trab,1,skewness)
Assimetria
Curtose <- apply(m_trab,1,kurtosis)
Curtose
Descriptores1 <- cbind(Sum, Mean, DP, Max, Min, S, E, H, D)
Descriptores1 <- as.data.frame(Descriptores1)
Descriptores1
#Descriptores1 <- Descriptores1 %>% rownames_to_column(var="Espécies") #da nome a primei
SomaTotalD <- apply(Descriptores1,2,sum)
SomaTotalD
MediaTotalD <- apply(Descriptores1,2,mean)
MediaTotalD
DPTotalD <- apply(Descriptores1,2,sd)
DPTotalD
Descriptores2 <- cbind(SomaTotalD, MediaTotalD, DPTotalD)
Descriptores2 <- as.data.frame(Descriptores2)
Descriptores2 <- t(Descriptores2)
Descriptores2
DescriptoresFinal <- rbind(Descriptores1, Descriptores2)
DescriptoresFinal
DescriptoresFinal <- round (DescriptoresFinal, 2)
DescriptoresFinal
#Fazendo uma tabela
library(gt)
df <- DescriptoresFinal
ncol(df); nrow(df) #no. de N colunas x M linhas
df <- cbind(Spp = rownames(df), df)
gt(df, rowname_col = "Espécie", caption = "Descriptores da diversidade por espécie (col
Normalidade1 <- cbind(Assimetria, Curtose)
Normalidade1 <- as.data.frame(Normalidade1)
Normalidade1
SomaTotalN <- apply(Normalidade1,2,sum)
SomaTotalN
MediaTotalN <- apply(Normalidade1,2,mean)
MediaTotalN
DPTotalN <- apply(Normalidade1,2,sd)
DPTotalN
Normalidade2<-cbind(SomaTotalN, MediaTotalN, DPTotalN)
Normalidade2<-as.data.frame(Normalidade2)
Normalidade2 <- t(Normalidade2) #"t" transpõe a matriz
Normalidade2
NormalidadeFinal <- rbind(Normalidade1, Normalidade2)

```

```

NormalidadeFinal
NormalidadeFinal <- round(NormalidadeFinal, 2)
NormalidadeFinal
#Fazendo uma tabela
nf <- NormalidadeFinal
ncol(nf); nrow(nf) #no. de N colunas x M linhas
nf <- cbind(Spp = rownames(nf), nf)
gt(nf, rowname_col = "Espécie", caption = "Descritores da normalidade por espécie (coluna)")
write.table(data.frame("Spp"=rownames(DescriidoresFinal),
                        DescriidoresFinal),
            "DescritoresSPP.txt",
            row.names=FALSE,
            sep="\t")
write.table(data.frame("Spp"=rownames(NormalidadeFinal),
                        NormalidadeFinal),
            "NormalidadeSPP.txt",
            row.names=FALSE,
            sep="\t")
m_trab <- ppbio
m_trab
Hill <- renyi(m_trab,
               scales = c(0:5),
               hill = TRUE)
Hill
library(ggplot2)
library(tidyr)
library(tidyverse)
grafico1 <- Hill %>%
  rownames_to_column() %>%
  pivot_longer(-rowname) %>%
  mutate(name = factor(name, name[1:length(Hill)])) %>%
  ggplot(aes(x = name, y = value, group = rowname,
             col = rowname)) +
  geom_point(size = 2) +
  geom_line(size = 1) +
  xlab("Parâmetro de ordem de diversidade (q)") +
  ylab("Diversidade") +
  labs(col = "Locais") +
  theme_bw() +
  theme(text = element_text(size = 5)) #ajustar a fonte caso nao caiba no output html
grafico1
ggsave(grafico1, dpi = 300, filename = "fig-hill.png")
abund <- colSums(m_trab)
#abund <- m_trab[1, ] #escolhe a primeira linha para a distribuição de abundância

```

```

df <- data.frame(sp = colnames(m_trab),
                 abun = abund)
grafico2 <- ggplot(df, aes(fct_reorder(sp, -abun),
                           abun, group = 1)) +
  geom_col() +
  geom_line(col = "red", linetype = "dashed") +
  geom_point(col = "red") +
  xlab("Espécies") +
  ylab("Abundância") +
  theme_bw() +
  theme(axis.text.x = element_text(
    angle = 45,
    hjust = 1,
    face = "italic"))
grafico2
ggsave(grafico2, dpi = 300, filename = "fig-abun.png")
m_trab_r <- mutate(m_trab, across(everything(), ceiling)) #arredonda a matriz original
colnames(t(m_trab_r))
rarefa <- t(m_trab_r[c("S-R-CT2", "S-R-CP2", "S-A-TA2", "B-A-MU2"),]) #curva de rarefação
class(m_trab)
## Curva de rarefação para todos os sítios
#rarefa <- t(m_trab_r)
## Curva de rarefação para as 8 UA's com maior soma
#m_trab_r <- as.data.frame(t(m_trab_r)) #transpõe a matriz
#col_sums <- colSums(m_trab_r)
#largest_columns <- names(sort(col_sums, decreasing = TRUE)[1:8])
#rarefa <- m_trab_r[largest_columns] #curva de rarefação para as 8 UA's com maior soma
library(iNEXT)
out <- iNEXT(rarefa, q = 0,
             datatype = "abundance",
             size = NULL,
             endpoint = 1500, #define o comprimento de eixo x
             knots = 40,
             se = TRUE,
             conf = 0.95,
             nboot = 50)
grafico3 <- ggiNEXT(out, type = 1, facet.var="None") +
  theme_bw() +
  labs(fill = "Áreas") +
  xlab("Número de indivíduos") +
  ylab("Riqueza de espécies") +
  theme(legend.title=element_blank()) #ver como fica com facet.var="Assemblage"
grafico3
ggsave(grafico3, dpi = 300, filename = "fig-rare1.png")

```

```

acumula <- specaccum(m_trab,
                      method = "random")
acumula
plot(acumula)
plot_data <- data.frame("UAs" = c(0, acumula$sites),
                        "Riqueza" = c(0, acumula$richness),
                        "lower" = c(0, acumula$richness - acumula$sd),
                        "upper" = c(0, acumula$richness + acumula$sd))
gLocais <- ggplot(plot_data, aes(x = UAs, y = Riqueza)) +
  geom_point(color = "blue", size = 4) +
  geom_line(color = "blue", lwd = 2) +
  geom_ribbon(aes(ymin = lower, ymax = upper),
              linetype=2, alpha=0.3, fill = "yellow") +
  ylab("Riqueza acumulada") +
  theme_classic() +
  theme(text = element_text(size = 16))
gLocais
plot_data <- data.frame("Individuos" = c(0, acumula$individuals),
                        "Riqueza" = c(0, acumula$richness),
                        "lower" = c(0, acumula$richness - acumula$sd),
                        "upper" = c(0, acumula$richness + acumula$sd))
gInd <- ggplot(plot_data, aes(x = Individuos, y = Riqueza)) +
  geom_point(color = "blue", size = 4) +
  geom_line(color = "blue", lwd = 2) +
  geom_ribbon(aes(ymin = lower, ymax = upper),
              linetype=2, alpha=0.3, fill = "yellow") +
  ylab("Riqueza acumulada") +
  theme_classic() +
  theme(text = element_text(size = 16))
gInd
#Matriz transposta
m_trab <- t(m_trab) #transpõe a matriz
Sum <- rowSums(m_trab)
Sum <- apply(m_trab,1,sum)
Mean <- rowMeans(m_trab)
Mean <- apply(m_trab,1,mean)
DP <- apply(m_trab,1,sd)
Max <- apply(m_trab,1,max)
Min <- apply(m_trab,1,min)
#library(vegan)
bin <- decostand(m_trab,"pa")
S <- apply(bin,1,sum)
Riqueza <- specnumber(m_trab)
Riqueza_total <- specnumber(colSums(m_trab))

```

```

H <- diversity(m_trab)
D <- diversity(m_trab, "simpson")
D[is.na(D)] <- 0 #substitui NA ou NaN por 0
E <- H/log(specnumber(m_trab))
E[is.na(E)] <- 0 #substitui NA ou NaN por 0
#library(moments)
Assimetria <- apply(m_trab,1,skewness)
Curtose <- apply(m_trab,1,kurtosis)
m_trab <- t(m_trab) #traz de volta
Descritores1 <- cbind(Sum, Mean, DP, Max, Min, S, E, H, D)
Descritores1 <- as.data.frame(Descritores1)
Descritores1
#Descritores1 <- Descritores1 %>% rownames_to_column(var="Espécies") #da nome a primeiros
SomaTotalD <- apply(Descritores1,2,sum)
SomaTotalD
MediaTotalD <- apply(Descritores1,2,mean)
MediaTotalD
DPTotalD <- apply(Descritores1,2,sd)
DPTotalD
Descritores2 <- cbind(SomaTotalD, MediaTotalD, DPTotalD)
Descritores2 <- as.data.frame(Descritores2)
Descritores2 <- t(Descritores2)
Descritores2
DescritoresFinal <- rbind(Descritores1, Descritores2)
DescritoresFinal
DescritoresFinal <- round (DescritoresFinal, 2)
#Fazendo uma tabela
#library(gt)
df <- DescritoresFinal
ncol(df); nrow(df) #no. de N colunas x M linhas
df <- cbind(UA = rownames(df), df)
gt(df, caption = "Descritores da diversidade por Unidade Amostral (UA). Sum, soma; mean, media; SD, desvio padrao")
Normalidade1 <- cbind(Assimetria, Curtose)
Normalidade1 <- as.data.frame(Normalidade1)
Normalidade1 <- na.omit(Normalidade1) #remove NA e NaN
Normalidade1
SomaTotalN <- apply(Normalidade1,2,sum)
SomaTotalN
MediaTotalN <- apply(Normalidade1,2,mean)
MediaTotalN
DPTotalN <- apply(Normalidade1,2,sd)
DPTotalN
Normalidade2<-cbind(SomaTotalN, MediaTotalN, DPTotalN)
Normalidade2<-as.data.frame(Normalidade2)

```

```
Normalidade2 <- t(Normalidade2) #"t" transpõe a matriz
Normalidade2
NormalidadeFinal <- rbind(Normalidade1, Normalidade2)
NormalidadeFinal
NormalidadeFinal <- round(NormalidadeFinal, 2)
#Fazendo uma tabela
nf <- NormalidadeFinal
ncol(nf); nrow(nf) #no. de N colunas x M linhas
nf <- cbind(UA = rownames(nf), nf)
gt(nf, caption = "Descritores da normalidade por Unidade Amostral (UA)")
write.table(data.frame("Spp"=rownames(DescritoresFinal),
                       DescritoresFinal),
            "DescritoresUA.csv",
            row.names=FALSE,
            sep="\t")
write.table(data.frame("Spp"=rownames(NormalidadeFinal),
                       NormalidadeFinal),
            "NormalidadeUA.txt",
            row.names=FALSE,
            sep="\t")
```

Referências

Chapter 12

R Modulo 4.2 - Decomposição da diversidade

RESUMO

Na ecologia de comunidades, a diversidade de espécies é uma medida importante para entender a complexidade e a estrutura de uma comunidade de organismos. Vamos explorar algumas das métricas comuns usadas para avaliar a diversidade de espécies.

Apresentação

Na ecologia de comunidades, a diversidade de espécies é uma medida importante para entender a complexidade e a estrutura de uma comunidade de organismos. Vamos explorar algumas das métricas comuns usadas para avaliar a diversidade de espécies. Essas métricas são usadas para avaliar a diversidade e a estrutura de comunidades ecológicas. Elas podem fornecer informações importantes sobre como as diferentes espécies interagem em um ecossistema e como a diversidade de espécies pode ser afetada por mudanças ambientais ou distúrbios (MAGUR-RAN, 1988).

12.1 Sobre os dados

Usaremos para esse tutorial dois conjuntos de dados. A **Matriz comunitária** (ppbio06c-peixes.xlsx) de dados coletados no Programa de Pesquisa em Biodiversidade - PPBio (Veja Programa de Pesquisa em Biodiversidade – PPBio). Esses são dados de espécies de peixes distribuídas em diversas unidades amostrais

(UA's ou sítios). Essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados para os valores de Captura Por Unidade de Esforço (CPUE), nem foram relativizados ou transformados.

Além disso usaremos a **tabela de agrupamentos** (ppbio06-grupos)

Revise as informações sobre as bases de dados no capítulo **Bases de Dados**. A matriz de dados para esse Módulo pode ser baixada na seção Arquivos disponíveis.

Revise o módulo anterior **Estrutura da Comunidade**

12.2 Organização básica

Apêndices

Sites consultados

Primeira coluna para nomes das linhas

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executa-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

Referências

Chapter 13

R Modulo 4.3 - Estrutura do habitat

RESUMO

Apresentação

13.1 Sobre os dados

Usaremos para esse tutorial dois conjuntos de dados. A **Matriz ambiental** (ppbio06p-amb.xlsx) de dados coletados no Programa de Pesquisa em Biodiversidade - PPBio (Veja Programa de Pesquisa em Biodiversidade – PPBio) (MEDEIROS; SILVA; RAMOS, 2008). Esses são dados de **variáveis ambientais** distribuídas em diversas unidades amostrais (UA's ou sítios). Essa é a **matriz bruta de dados**, porque os valores ainda não foram transformados (SOKAL; ROHLF, 1995). Note que as variáveis foram medidas em diferentes unidades (cm, m, °C, mg/L, etc.), com uma alta amplitude de variação, sugerido uso de matriz transformada e/ou reescalada. Além disso usaremos a **tabela de agrupamentos** (ppbio06-grupos).

Revise as informações sobre as bases de dados no Capítulo 3. As bases de dados para esse Módulo podem ser baixadas na Seção 3.2.

13.2 Organização básica

```
dev.off() #apaga os graficos, se houver algum
rm(list=ls(all=TRUE)) #limpa a memória
cat("\014") #limpa o console
```

13.2.1 Instalando os pacotes necessários para esse módulo.

Se você já fez o módulo anterior, não vai precisar instalar nenhum pacote novo. Os pacotes necessários serão carregados ao longo desse módulo.

```
install.packages("psych")
install.packages("corrplot")
```

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

13.3 Importando a planilha

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações.

- Ajuste a primeira linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.
- Ajuste o parâmetro `sheet = "Sheet1"` para refletir a aba correta do arquivo .xlsx a ser importado. Vamos trabalhar com os dados do ano 1.

```
#dir <- getwd() #criamos um vetor com o diretório de trabalho
#shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
library(openxlsx)
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06p-amb
                    rowNames = T, colNames = T,
                    sheet = "ano1")
str(ppbio)
#View(ppbio)
ppbio[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.

## 'data.frame':   23 obs. of  30 variables:
## $ h.macrof  : num  8.33 0 5.83 0 0 ...
## $ h.grass   : num  20 23.3 0 20 0 ...
## $ h.subveg  : num  10 3 16.7 0 0 ...
## $ h.overhveg: num  0 26.7 33.3 0 33.3 ...
## $ h.litter  : num  0.667 2.333 23.667 0 1 ...
## $ h.filalgae: num  0 3.33 25 0 0 ...
```

```

## $ h.attalgae : num 12 0.667 3.333 1 0 ...
## $ h.roots : num 0 3.33 0 0 5 ...
## $ h.lrgdeb : num 1.67 1.67 0 0 3.33 ...
## $ h.smldeb : num 1.67 2 10 0 7 ...
## $ s.mud : num 65 16.667 81.667 40 0.667 ...
## $ s.sand : num 30 70 8.33 40 87.67 ...
## $ s.smlgrav : num 0 5 1.67 0 3.33 ...
## $ s.lrggrav : num 0 5 0 0 3.33 ...
## $ s.cobbles : num 1.67 3.33 0 20 5 ...
## $ s.rocks : num 0 0 0 0 0 ...
## $ s.bedrock : num 3.33 0 8.33 0 0 ...
## $ m.elev : num 226 169 270 226 169 270 226 169 270 226 ...
## $ m.river : num 163 83 110 163 83 ...
## $ m.stream : num 163.2 13.6 37.1 163.2 13.6 ...
## $ m.distsource: num 84.45 6.15 8.85 84.45 6.15 ...
## $ m.distmouth : num 217 186 255 217 186 ...
## $ m.maxslope : num 30 60 60 30 60 60 30 60 60 30 ...
## $ m.maxdepth : num 106 60 154 105 68 118 110 79 109 74 ...
## $ m.habdepth : num 81.3 22.7 54.7 67 32.7 ...
## $ m.width : num 19.6 17.2 102 16.1 18.5 ...
## $ a.veloc : num 0.167 0.159 0.1 0.125 0 ...
## $ a.temp : num 32.9 35.2 34 32 29 ...
## $ a.do : num 6.51 6.86 4.82 5.38 3.02 ...
## $ a.transp : num 46 26 61 44 33 ...
##           h.macroh h.grass h.subveg h.overhveg h.litter
## S-R-CT1 8.333333 20.00000 10.00000 0.00000 0.6666667
## S-R-CP1 0.000000 23.33333 3.00000 26.66667 2.3333333
## S-A-TA1 5.833333 0.00000 16.66667 33.33333 23.6666660
## S-R-CT2 0.000000 20.00000 0.00000 0.00000 0.0000000
## S-R-CP2 0.000000 0.00000 0.00000 33.33333 1.0000000

```

Exibindo os dados importados (esses comando são “case-sensitive” `ignore.case(object)`).

```

#View(ppbio)
print(ppbio[1:8,1:8])
ppbio[1:10,1:10]
str(ppbio)
mode(ppbio)
class(ppbio)

```

13.4 Reset point

```

m_amb <- ppbio
#pat <- "^\$"

```

```
#m_amb <- m_amb[!grepl(pat, rownames(m_amb)), ] #exclui quem começa com pat
```

Aqui cria-se um novo objeto do R (`m_trab`, ou a matriz de trabalho, para esse momento) que substitui a matriz de dados original, por uma nova matriz que pode ser a matriz relativizada, transformada, transposta, etc. Dessa forma, mantemos a matriz de dados original caso precisemos dela novamente (Veja a Tabela ??).

Revise Seção 3.2.2 e a Tabela de Abreviações (@ref(tab:200m_2)) na mesma Seção que resumem os tipos de matrizes e suas abreviações, para os nossos códigos.

13.5 Correlograma e remoção de variáveis redundantes ou desnecessárias

```
library(psych)
colnames(m_amb)
pairs.panels(m_amb[,1:10],
             method = "pearson", # correlation method
             scale = FALSE, lm = FALSE,
             hist.col = "#00AFBB", pch = 19,
             density = TRUE, # show density plots
             ellipses = TRUE, # show correlation ellipses
             alpha = 0.5)

cor <- cor(m_amb)
cor
library(corrplot)
corrplot(cor, method = "circle")
#win.print()
#corrplot(cor, method = "circle")
#dev.off()

## [1] "h.macroh"      "h.grass"       "h.subveg"      "h.overhveg"    "h.litter"
## [6] "h.filalgae"    "h.attalgae"    "h.roots"       "h.lrgdeb"     "h.smldeb"
## [11] "s.mud"         "s.sand"        "s.smlgrav"    "s.lrggrav"    "s.cobbles"
## [16] "s.rocks"       "s.bedrock"     "m.elev"        "m.river"      "m.stream"
## [21] "m.distsource"  "m.distmouth"   "m.maxslope"   "m.maxdepth"   "m.habdepth"
## [26] "m.width"        "a.veloc"       "a.temp"        "a.do"         "a.transp"
##           h.macroh      h.grass      h.subveg      h.overhveg      h.litter
## h.macroh      1.00000000 -0.246531504 -0.13690655 -0.0460990189 -0.062637491
## h.grass      -0.24653150  1.000000000  0.48407509 -0.1460764735 -0.001721976
## h.subveg     -0.13690655  0.484075094  1.00000000  0.0635287907  0.399197150
```

13.5. CORRELOGRAMA E REMOÇÃO DE VARIÁVEIS REDUNDANTES OU DESNECESSÁRIAS 273

```

## h.overhveg -0.04609902 -0.146076474 0.06352879 1.0000000000 0.603643875
## h.litter -0.06263749 -0.001721976 0.39919715 0.6036438749 1.0000000000
## h.filalgae 0.30739321 -0.225737505 -0.06840003 0.3151725645 0.391954687
## h.attalgae -0.20061743 -0.203856535 -0.15804953 -0.0588612264 -0.074417928
## h.roots -0.17114523 -0.042846121 -0.12422422 0.7224571129 -0.020670464
## h.lrgdeb 0.03975689 0.517038783 -0.13497524 -0.0002521598 -0.076622164
## h.smldeb -0.11282416 0.430029457 0.32450367 0.3420275576 0.524639005
## s.mud -0.02825772 -0.213876066 0.09356431 -0.3318047182 0.081241055
## s.sand 0.10165279 0.290788117 -0.02468805 0.2908755666 -0.108025446
## s.smlgrav 0.02303766 -0.230252599 -0.26591389 0.1532179364 0.008573628
## s.lrggrav 0.05597825 -0.088160779 -0.19524294 0.1761326511 -0.054082053
## s.cobbles -0.24512665 -0.007205913 -0.15483540 0.0694742877 -0.093430622
## s.rocks -0.23075335 -0.259155527 -0.21954489 -0.0873692368 -0.121239165
## s.bedrock -0.03652910 -0.083573123 0.30828355 0.5073656218 0.900953133
## m.elev -0.08310889 0.403591279 0.35058982 -0.4003114855 -0.131742426
## m.river -0.23257092 0.335464812 0.24926832 -0.5955683647 -0.242615414
## m.stream -0.06608660 -0.004596811 -0.07497728 -0.2227767752 -0.096058730
## m.distsource -0.16898179 0.005784151 -0.10079544 -0.2397842969 -0.143925113
## m.distmouth -0.09461959 0.058821726 -0.01701650 -0.2886438017 0.009362746
## m.maxslope 0.06917689 -0.432093408 -0.28679549 0.2396730909 0.112055941
## m.maxdepth 0.03186630 0.402241943 0.19851159 -0.0016801722 0.489997569
## m.habdepth -0.18039247 -0.045055633 -0.13643528 0.0703614430 0.206858486
## m.width 0.10609087 0.339661628 0.37489017 -0.2460756693 -0.036647042
## a.veloc -0.13503225 0.204580859 0.11506415 0.3775858154 0.314867909
## a.temp 0.18273224 0.093689794 0.10708693 0.3872777253 0.376177481
## a.do 0.26259801 -0.202794752 -0.60025144 -0.2351348072 -0.186841357
## a.transp 0.30879962 -0.007818165 0.06381010 -0.0405424177 0.149062804
##          h.filalgae h.attalgae h.roots h.lrgdeb h.smldeb
## h.macroph 0.307393212 -0.2006174303 -0.17114523 0.0397568923 -0.1128241586
## h.grass -0.225737505 -0.2038565351 -0.04284612 0.5170387832 0.4300294573
## h.subveg -0.068400029 -0.1580495292 -0.12422422 -0.1349752429 0.3245036701
## h.overhveg 0.315172564 -0.0588612264 0.72245711 -0.0002521598 0.3420275576
## h.litter 0.391954687 -0.0744179283 -0.02067046 -0.0766221642 0.5246390055
## h.filalgae 1.0000000000 0.1650218328 -0.12170792 0.0330671467 0.0566631854
## h.attalgae 0.165021833 1.0000000000 -0.15141029 -0.1110887477 -0.0007226332
## h.roots -0.121707915 -0.1514102880 1.00000000 0.0359061688 0.1278778876
## h.lrgdeb 0.033067147 -0.1110887477 0.03590617 1.0000000000 0.5126858957
## h.smldeb 0.056663185 -0.0007226332 0.12787789 0.5126858957 1.0000000000
## s.mud -0.203493831 0.0966629147 -0.45742597 -0.2661041300 -0.1358265852
## s.sand 0.131463557 -0.2065073854 0.42226823 0.3631216979 0.2116433147
## s.smlgrav 0.056355761 -0.2466165417 0.37504581 -0.2178131916 -0.1938969370
## s.lrggrav 0.071026543 -0.2019284184 0.45157180 -0.1064395977 -0.1131217094
## s.cobbles 0.193135679 0.4075488652 0.08813464 -0.1218105205 -0.2307397229
## s.rocks 0.144067566 0.5595435077 -0.15324139 -0.1450923302 -0.0896567456
## s.bedrock 0.338537041 -0.0146391847 -0.09492929 -0.0956520825 0.3503440506
## m.elev -0.203448440 0.0716487680 -0.36640440 0.2656332571 0.1727891903

```

```

## m.river      -0.405669260 -0.0195694902 -0.51853032  0.1309548871 -0.0189887633
## m.stream     -0.185417996 -0.0598126521 -0.21720079 -0.1720191090 -0.1810175388
## m.distsource -0.254533090 -0.0572352576 -0.18429493 -0.1776252995 -0.2083808690
## m.distmouth   -0.217780469 -0.2330343386 -0.33511609  0.1038897654  0.0767920775
## m.maxslope    0.145284072 -0.1601113205  0.18673984 -0.1907894213 -0.1808970343
## m.maxdepth    0.114574709 -0.2474674119 -0.35179352  0.5122182964  0.6060301921
## m.habdepth    0.002882856  0.0164745166 -0.09468290  0.0036984721  0.0212264396
## m.width       -0.011447418  0.0797573366 -0.28844323  0.2562191774  0.1713597968
## a.veloc       -0.002302649 -0.0605797247  0.20772262 -0.1007706522 -0.0883913083
## a.temp        -0.047385170 -0.3528557182  0.21266182 -0.0633115802  0.0495255921
## a.do          0.075932799  0.2107159895 -0.19965987 -0.0527554510 -0.2305667359
## a.transp      0.486443594  0.1991293428 -0.26299621  0.2727840805  0.2437019444
##           s.mud    s.sand    s.smlgrav  s.lrggrav  s.cobbles
## h.macroph    -0.02825772  0.10165279  0.023037659  0.05597825 -0.245126650
## h.grass       -0.21387607  0.29078812 -0.230252599 -0.08816078 -0.007205913
## h.subveg      0.09356431 -0.02468805 -0.265913890 -0.19524294 -0.154835404
## h.overhveg   -0.33180472  0.29087557  0.153217936  0.17613265  0.069474288
## h.litter      0.08124106 -0.10802545  0.008573628 -0.05408205 -0.093430622
## h.filalgae   -0.20349383  0.13146356  0.056355761  0.07102654  0.193135679
## h.attalgae    0.09666291 -0.20650739 -0.246616542 -0.20192842  0.407548865
## h.roots       -0.45742597  0.42226823  0.375045814  0.45157180  0.088134635
## h.lrgdeb      -0.26610413  0.36312170 -0.217813192 -0.10643960 -0.121810520
## h.smldeb      -0.13582659  0.21164331 -0.193896937 -0.11312171 -0.230739723
## s.mud         1.00000000 -0.95732831 -0.196136235 -0.20727196 -0.436951580
## s.sand        -0.95732831  1.00000000  0.077601964  0.06113988  0.208220232
## s.smlgrav     -0.19613624  0.07760196  1.000000000  0.91794933 -0.018580075
## s.lrggrav     -0.20727196  0.06113988  0.917949325  1.000000000  0.092736437
## s.cobbles     -0.43695158  0.20822023 -0.018580075  0.09273644  1.000000000
## s.rocks       -0.05450604 -0.08712170 -0.290187186 -0.20917141  0.471919983
## s.bedrock     0.20788751 -0.23158686 -0.068825165 -0.12957657 -0.117937719
## m.elev        0.46647761 -0.31506556 -0.363536747 -0.35844427 -0.418891510
## m.river       0.52267706 -0.35870643 -0.487625688 -0.54998626 -0.404162728
## m.stream      0.25913400 -0.29280195 -0.351718673 -0.23344861  0.162336709
## m.distsource  0.20798419 -0.24660914 -0.335764568 -0.23558448  0.197820570
## m.distmouth   -0.12783195  0.26298990 -0.055199753 -0.33221276 -0.343006017
## m.maxslope    -0.48822579  0.47051421  0.480231912  0.19989479  0.003285850
## m.maxdepth    -0.08984828  0.15690152 -0.233610098 -0.23813743 -0.181918712
## m.habdepth    -0.29918013  0.24152725 -0.124634583 -0.20123580  0.329637349
## m.width       0.49913277 -0.37568825 -0.310246711 -0.23740245 -0.381602616
## a.veloc       -0.07881376  0.02561095 -0.011480476  0.04477510  0.201826380
## a.temp        -0.20642510  0.16196759  0.223976700  0.27319038  0.054683155
## a.do          0.29936318 -0.31184728  0.228246923  0.14674375 -0.188261015
## a.transp      -0.07274786  0.08367750 -0.090575822 -0.07405382  0.010898759
##           s.rocks  s.bedrock  m.elev   m.river   m.stream
## h.macroph   -0.230753352 -0.03652910 -0.08310889 -0.23257092 -0.066086599
## h.grass     -0.259155527 -0.08357312  0.40359128  0.33546481 -0.004596811

```

13.5. CORRELOGRAMA E REMOÇÃO DE VARIÁVEIS REDUNDANTES OU DESNECESSÁRIAS 275

```

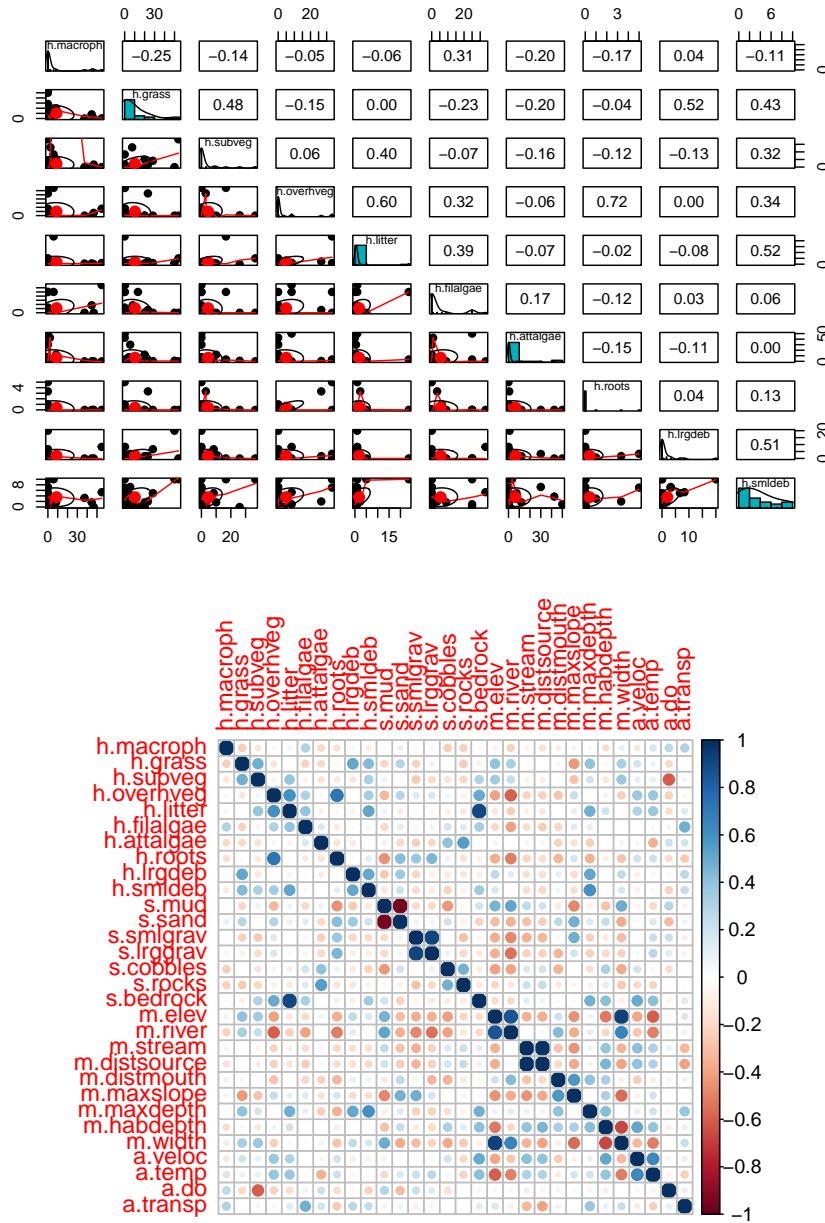
## h.subveg      -0.219544891  0.30828355  0.35058982  0.24926832 -0.074977285
## h.overhveg   -0.087369237  0.50736562 -0.40031149 -0.59556836 -0.222776775
## h.litter     -0.121239165  0.90095313 -0.13174243 -0.24261541 -0.096058730
## h.filalgae   0.144067566  0.33853704 -0.20344844 -0.40566926 -0.185417996
## h.attalgae   0.559543508  -0.01463918  0.07164877 -0.01956949 -0.059812652
## h.roots      -0.153241391  -0.09492929 -0.36640440 -0.51853032 -0.217200787
## h.lrgdeb     -0.145092330  -0.09565208  0.26563326  0.13095489 -0.172019109
## h.smldeb     -0.089656746  0.35034405  0.17278919 -0.01898876 -0.181017539
## s.mud        -0.054506037  0.20788751  0.46647761  0.52267706  0.259134004
## s.sand       -0.087121705  -0.23158686 -0.31506556 -0.35870643 -0.292801947
## s.smlgrav    -0.290187186  -0.06882517 -0.36353675 -0.48762569 -0.351718673
## s.lrggrav    -0.209171407  -0.12957657 -0.35844427 -0.54998626 -0.233448612
## s.cobbles    0.471919983  -0.11793772 -0.41889151 -0.40416273  0.162336709
## s.rocks      1.000000000  -0.12799218 -0.21907348 -0.13545807  0.215737335
## s.bedrock    -0.127992185  1.00000000  -0.19742578 -0.20071090  0.131806173
## m.elev       -0.219073482  -0.19742578  1.00000000  0.85251041 -0.351025053
## m.river      -0.135458068  -0.20071090  0.85251041  1.00000000  0.017040248
## m.stream     0.215737335  0.13180617 -0.35102505  0.01704025  1.000000000
## m.distsource 0.248398309  0.08398842 -0.36628911  0.04500187  0.987553629
## m.distmouth  -0.128592531  -0.03694612  0.21088433  0.43005863 -0.285305822
## m.maxslope   -0.063334534  0.04208746 -0.45271956 -0.37494422 -0.445502469
## m.maxdepth   -0.078095404  0.47564528  0.08022982  0.12595144  0.094085376
## m.habdepth   0.133210844  0.41315901 -0.52848500 -0.21000817  0.395997655
## m.width      -0.267413674  -0.09980488  0.92959448  0.66160089 -0.341120113
## a.veloc      -0.203924152  0.50163597 -0.39264584 -0.27533927  0.410516289
## a.temp       -0.221454717  0.41538130 -0.60522317 -0.50743529  0.318565681
## a.do         0.004835464  -0.06655234 -0.04245875 -0.05281226  0.104875559
## a.transp    -0.016360295  0.14038232  0.16444871 -0.08762683 -0.339047044
## m.distsource m.distmouth  m.maxslope  m.maxdepth  m.habdepth
## h.macroph   -0.168981791  -0.094619591  0.069176887  0.031866297 -0.180392470
## h.grass      0.005784151  0.058821726 -0.432093408  0.402241943 -0.045055633
## h.subveg     -0.100795436  -0.017016500 -0.286795491  0.198511591 -0.136435275
## h.overhveg   -0.239784297  -0.288643802  0.239673091 -0.001680172  0.070361443
## h.litter     -0.143925113  0.009362746  0.112055941  0.489997569  0.206858486
## h.filalgae   -0.254533090  -0.217780469  0.145284072  0.114574709  0.002882856
## h.attalgae   -0.057235258  -0.233034339 -0.160111320 -0.247467412  0.016474517
## h.roots      -0.184294932  -0.335116093  0.186739840 -0.351793522 -0.094682904
## h.lrgdeb     -0.177625299  0.103889765 -0.190789421  0.512218296  0.003698472
## h.smldeb     -0.208380869  0.076792078 -0.180897034  0.606030192  0.021226440
## s.mud        0.207984189  -0.127831955 -0.488225793 -0.089848279 -0.299180125
## s.sand       -0.246609145  0.262989896  0.470514210  0.156901516  0.241527247
## s.smlgrav    -0.335764568  -0.055199753  0.480231912 -0.233610098 -0.124634583
## s.lrggrav    -0.235584477  -0.332212758  0.199894791 -0.238137430 -0.201235803
## s.cobbles    0.197820570  -0.343006017  0.003285850 -0.181918712  0.329637349
## s.rocks      0.248398309  -0.128592531 -0.063334534 -0.078095404  0.133210844
## s.bedrock    0.083988424  -0.036946118  0.042087459  0.475645280  0.413159010

```

```

## m.elev      -0.366289105  0.210884333 -0.452719558  0.080229817 -0.528485000
## m.river     0.045001873  0.430058635 -0.374944223  0.125951443 -0.210008174
## m.stream    0.987553629 -0.285305822 -0.445502469  0.094085376  0.395997655
## m.distsource 1.000000000 -0.208886217 -0.380287859  0.070018120  0.454126171
## m.distmouth -0.208886217  1.000000000  0.575655522  0.409794260  0.329568908
## m.maxslope   -0.380287859  0.575655522  1.000000000 -0.033910389  0.326367540
## m.maxdepth   0.070018120  0.409794260 -0.033910389  1.000000000  0.425648040
## m.habdepth   0.454126171  0.329568908  0.326367540  0.425648040  1.000000000
## m.width      -0.403406954 -0.091476026 -0.569731109  0.010919605 -0.663930169
## a.veloc      0.409162331 -0.251728578 -0.087772483  0.109911454  0.516555276
## a.temp       0.303332037 -0.107191864  0.190578794  0.204458211  0.390454228
## a.do         0.054863239 -0.066103944 -0.039194058 -0.068299977 -0.098869325
## a.transp    -0.402578165  0.084213818  0.008475289  0.389325169  0.109987158
##           m.width      a.veloc      a.temp      a.do      a.transp
## h.macroph   0.106090873 -0.135032245  0.18273224  0.262598008  0.308799625
## h.grass     0.339661628  0.204580859  0.09368979 -0.202794752 -0.007818165
## h.subveg    0.374890169  0.115064150  0.10708693 -0.600251445  0.063810096
## h.overhveg  -0.246075669  0.377585815  0.38727773 -0.235134807 -0.040542418
## h.litter    -0.036647042  0.314867909  0.37617748 -0.186841357  0.149062804
## h.filalgae  -0.011447418 -0.002302649 -0.04738517  0.075932799  0.486443594
## h.attalgae   0.079757337 -0.060579725 -0.35285572  0.210715990  0.199129343
## h.roots     -0.288443233  0.207722616  0.21266182 -0.199659873 -0.262996213
## h.lrgdeb    0.256219177 -0.100770652 -0.06331158 -0.052755451  0.272784080
## h.smldeb    0.171359797 -0.088391308  0.04952559 -0.230566736  0.243701944
## s.mud        0.499132769 -0.078813761 -0.20642510  0.299363179 -0.072747855
## s.sand       -0.375688250  0.025610951  0.16196759 -0.311847278  0.083677500
## s.smlgrav   -0.310246711 -0.011480476  0.22397670  0.228246923 -0.090575822
## s.lrggrav   -0.237402453  0.044775100  0.27319038  0.146743753 -0.074053824
## s.cobbles    -0.381602616  0.201826380  0.05468315 -0.188261015  0.010898759
## s.rocks     -0.267413674 -0.203924152 -0.22145472  0.004835464 -0.016360295
## s.bedrock   -0.099804876  0.501635966  0.41538130 -0.066552340  0.140382315
## m.elev      0.929594477 -0.392645839 -0.60522317 -0.042458750  0.164448713
## m.river     0.661600887 -0.275339274 -0.50743529 -0.052812259 -0.087626835
## m.stream    -0.341120113  0.410516289  0.31856568  0.104875559 -0.339047044
## m.distsource -0.403406954  0.409162331  0.30333204  0.054863239 -0.402578165
## m.distmouth -0.091476026 -0.251728578 -0.10719186 -0.066103944  0.084213818
## m.maxslope   -0.569731109 -0.087772483  0.19057879 -0.039194058  0.008475289
## m.maxdepth   0.010919605  0.109911454  0.20445821 -0.068299977  0.389325169
## m.habdepth   -0.663930169  0.516555276  0.39045423 -0.098869325  0.109987158
## m.width      1.000000000 -0.326799203 -0.51641177 -0.002086238  0.232399977
## a.veloc      -0.326799203  1.000000000  0.64196801  0.063666386 -0.111321669
## a.temp       -0.516411771  0.641968008  1.000000000 -0.072530252 -0.241797672
## a.do         -0.002086238  0.063666386 -0.07253025  1.000000000  0.109896052
## a.transp    0.232399977 -0.111321669 -0.241797677  0.109896052  1.000000000

```



13.6 Deletando variáveis colineares

Primeiro vamos listar as variáveis com correlação maior que 70% ou $r > 0.7$.

```

# Get upper triangle (to avoid duplicate pairs)
r <- 0.7
cor_pairs <- which(abs(cor) > r & abs(cor) < 1, arr.ind = TRUE)

# Extract unique variable pairs
results <- data.frame(
  var1 = rownames(cor)[cor_pairs[, 1]],
  var2 = colnames(cor)[cor_pairs[, 2]],
  correlation = cor[cor_pairs]
)

# Remove duplicates (e.g. A-B and B-A)
results <- results[results$var1 < results$var2, ]

# Sort by absolute correlation (descending)
results <- results[order(-abs(results$correlation)), ]
results

##           var1      var2 correlation
## 12 m.distsource m.stream   0.9875536
## 5    s.mud       s.sand   -0.9573283
## 14   m.elev      m.width   0.9295945
## 6    s.lrggrav  s.smlgrav  0.9179493
## 8    h.litter    s.bedrock  0.9009531
## 11   m.elev      m.river   0.8525104
## 3    h.overhveg  h.roots   0.7224571

```

Agora que temos as variáveis correlacionadas, prosseguimos para deletá-las.

```

colnames(m_amb)
del_cols <- c("m.distsource", "m.elev") #exemplo, "g.river_length", "g.altitude"
m_amb_part <- m_amb[, !(colnames(m_amb) %in% del_cols)]

## [1] "h.macroph"     "h.grass"        "h.subveg"        "h.overhveg"      "h.litter"
## [6] "h.filalgae"    "h.attalgae"     "h.roots"        "h.lrgdeb"       "h.smldeb"
## [11] "s.mud"          "s.sand"         "s.smlgrav"      "s.lrggrav"      "s.cobbles"
## [16] "s.rocks"        "s.bedrock"      "m.elev"         "m.river"        "m.stream"
## [21] "m.distsource"   "m.distmouth"   "m.maxslope"     "m.maxdepth"    "m.habdepth"
## [26] "m.width"        "a.veloc"        "a.temp"         "a.do"          "a.transp"

```

13.7 Somando variáveis redundantes

Algumas dessas variáveis são redundantes, vamos soma-las.

```

m_amb_part$s.gravel <- m_amb_part$s.smlgrav + m_amb_part$s.lrggrav + m_amb_part$s.cobbles
m_amb_part <- m_amb_part[, !(colnames(m_amb_part)
                           %in% c("s.smlgrav", "s.lrggrav", "s.cobbles"))]
m_amb_part$s.rock <- m_amb_part$s.rocks + m_amb_part$s.bedrock
m_amb_part <- m_amb_part[, !(colnames(m_amb_part)
                           %in% c("s.rocks", "s.bedrock"))]
m_amb_part$h.algae <- m_amb_part$h.filalgae + m_amb_part$h.attalgae
m_amb_part <- m_amb_part[, !(colnames(m_amb_part)
                           %in% c("h.filalgae", "h.attalgae"))]
m_amb_part$h.debris <- m_amb_part$h.smldeb + m_amb_part$h.lrgdeb
m_amb_part <- m_amb_part[, !(colnames(m_amb_part)
                           %in% c("h.smldeb", "h.lrgdeb"))]

colnames(m_amb_part)
m_amb_part

## [1] "h.macroph"      "h.grass"        "h.subveg"       "h.overhveg"     "h.litter"
## [6] "h.roots"         "s.mud"          "s.sand"         "m.river"        "m.stream"
## [11] "m.distmouth"     "m.maxslope"     "m.maxdepth"    "m.habdepth"    "m.width"
## [16] "a.veloc"         "a.temp"         "a.do"          "a.transp"      "s.gravel"
## [21] "s.rock"          "h.algae"        "h.debris"
##           h.macroph   h.grass   h.subveg h.overhveg   h.litter   h.roots
## S-R-CT1  8.333333 20.000000 10.000000  0.000000  0.6666667 0.000000
## S-R-CP1  0.000000 23.333340  3.000000  26.666666  2.3333333 3.333333
## S-A-TA1  5.833333  0.0000000 16.666666  33.333332 23.6666660 0.000000
## S-R-CT2  0.000000 20.0000000  0.000000  0.000000  0.0000000 0.000000
## S-R-CP2  0.000000  0.0000000  0.000000  33.333332 1.0000000 5.000000
## S-A-TA2  54.833332  0.3333333  0.000000  8.333333  1.0000000 0.000000
## S-R-CT3  0.000000  0.1111111  0.000000  0.000000  0.0000000 0.000000
## S-R-CP3  0.000000  0.0000000  0.000000  0.000000  1.5000000 1.000000
## S-A-TA3  44.629631  8.1111107  0.000000  0.000000  0.6666667 0.000000
## S-R-CT4  0.000000  4.1666665  0.000000  0.000000  0.8333333 0.000000
## S-R-CP4  0.000000  0.0000000  0.000000  8.333333  1.0000000 0.000000
## S-A-TA4  37.309525  0.0000000  0.000000  0.000000  0.5714286 0.000000
## B-A-MU1  0.000000  50.0000000  0.000000  0.000000  1.0000000 0.000000
## B-A-GU1  46.666668  0.0000000  3.333333  3.333333  0.0000000 0.000000
## B-R-PC2  0.000000  2.0000000  0.000000  0.000000  0.4000000 0.000000
## B-A-MU2  0.000000  54.0000000  36.599998  0.000000  5.0000000 0.000000
## B-A-GU2  2.600000  7.5000000  26.000000  0.400000  0.0000000 0.000000
## B-R-PC3  0.000000  0.0000000  0.000000  0.000000  0.3333333 0.000000
## B-A-MU3  0.000000  28.3333340  8.333333  0.000000  1.0000000 0.000000
## B-A-GU3  0.000000  13.3333330  0.000000  0.000000  0.0000000 0.000000
## B-R-PC4  0.000000  0.0000000  0.000000  0.000000  0.5000000 0.000000
## B-A-MU4  0.000000  5.5555553  0.000000  0.000000  0.4444444 0.000000

```

```

## B-A-GU4  2.083333  0.0000000  0.000000  0.000000  0.0000000 0.000000
##           s.mud    s.sand   m.river  m.stream m.distmouth m.maxslope
## S-R-CT1  65.0000000 30.000000 163.2000 163.20000 217.1500      30
## S-R-CP1  16.6666660 70.000000 83.0500 13.55000 185.5500      60
## S-A-TA1  81.6666641 8.333333 110.2000 37.10000 254.5500      60
## S-R-CT2  40.0000000 40.000000 163.2000 163.20000 217.1500      30
## S-R-CP2  0.6666667 87.666664 83.0500 13.55000 185.5500      60
## S-A-TA2  5.0000000 95.000000 110.2000 37.10000 254.5500      60
## S-R-CT3  65.5555573 23.111111 163.2000 163.20000 217.1500      30
## S-R-CP3  48.7500000 22.500000 83.0500 13.55000 185.5500      60
## S-A-TA3  46.6666679 40.000000 110.2000 37.10000 254.5500      60
## S-R-CT4  95.0000000 1.833333 163.2000 163.20000 217.1500      30
## S-R-CP4  5.0000000 60.000000 83.0500 13.55000 185.5500      60
## S-A-TA4  59.1428566 38.714287 110.2000 37.10000 254.5500      60
## B-A-MU1  33.3333321 65.000000 214.0200 19.83333 290.8767      30
## B-A-GU1  96.6666641 3.333333 212.6667 32.33333 203.8333      30
## B-R-PC2  39.0000000 56.000000 196.8333 13.75000 401.6667      90
## B-A-MU2  20.6000004 77.000000 214.0200 19.83333 290.8767      30
## B-A-GU2  98.0000000 2.000000 212.6667 32.33333 203.8333      30
## B-R-PC3  33.6666679 63.000000 196.8333 13.75000 401.6667      90
## B-A-MU3  65.0000000 35.000000 214.0200 19.83333 290.8767      30
## B-A-GU3  87.7777786 6.666667 212.6667 32.33333 203.8333      30
## B-R-PC4  48.8750000 47.875000 196.8333 13.75000 401.6667      90
## B-A-MU4  91.7777786 3.222222 214.0200 19.83333 290.8767      30
## B-A-GU4  95.1666641 3.416667 212.6667 32.33333 203.8333      30
##           m.maxdepth m.habdepth m.width   a.veloc   a.temp     a.do a.transp
## S-R-CT1          106  81.333333  19.64 0.1666667 32.90000 6.510000 46.00000
## S-R-CP1          60  22.666667  17.24 0.1591512 35.20000 6.863333 26.00000
## S-A-TA1          154  54.666667 102.00 0.1000000 34.00000 4.820000 61.00000
## S-R-CT2          105  67.000000  16.10 0.1250000 32.00000 5.375000 44.00000
## S-R-CP2          68  32.666667  18.47 0.0000000 29.00000 3.015000 33.00000
## S-A-TA2          118  37.666667 100.00 0.0000000 29.00000 5.000000 90.00000
## S-R-CT3          110  32.333333   6.20 0.0000000 28.26667 6.000000 17.33333
## S-R-CP3          79  32.666667  15.10 0.0000000 29.66667 5.000000 50.33333
## S-A-TA3          109  22.333333  88.00 0.0000000 34.00000 9.000000 51.66667
## S-R-CT4          74  32.333333   5.40 0.0000000 32.60000 6.000000 16.00000
## S-R-CP4          64  45.333333  10.70 0.0000000 27.60000 4.900000 60.00000
## S-A-TA4          87  30.333333  72.20 0.0000000 29.53333 9.433333 67.00000
## B-A-MU1          152  32.666667 270.00 0.0000000 29.83333 5.666667 48.00000
## B-A-GU1          60   8.166667 330.00 0.0000000 29.23333 5.136667 25.66667
## B-R-PC2          110  49.333333  29.60 0.0000000 29.00000 5.635000 50.00000
## B-A-MU2          115  22.000000 247.63 0.0000000 29.00000 1.815000 43.00000
## B-A-GU2          69   7.000000 321.00 0.0000000 29.00000 1.850000 55.00000
## B-R-PC3          80  50.000000  27.30 0.0000000 29.00000 5.000000 32.33333
## B-A-MU3          117  25.666667 234.53 0.0000000 26.00000 5.700000 63.00000
## B-A-GU3          68   6.833333 314.20 0.0000000 24.00000 8.800000 51.66667

```

```

## B-R-PC4      95  52.833333  20.00 0.0000000 28.85000 5.100000 30.00000
## B-A-MU4     112  41.333333 239.00 0.0000000 25.95000 7.300000 89.00000
## B-A-GU4      63   4.666667 289.50 0.0000000 24.70000 8.750000 36.00000
##          s.gravel    s.rock   h.algae   h.debris
## S-R-CT1  1.6666666  3.333333 12.0000000 3.3333333
## S-R-CP1 13.3333333  0.000000  3.9999999 3.6666666
## S-A-TA1  1.6666666  8.333333 28.3333333 10.0000000
## S-R-CT2 20.0000000  0.000000  1.0000000 0.0000000
## S-R-CP2 11.6666665  0.000000  0.0000000 10.3333333
## S-A-TA2  0.0000000  0.000000 33.3333321 13.6666665
## S-R-CT3  3.0000000  8.333333  0.0000000 2.2222223
## S-R-CP3 28.7500000  0.000000  6.7500000 3.6666667
## S-A-TA3 13.3333330  0.000000  9.0000000 1.6666666
## S-R-CT4  0.3333333  2.833333  7.5000000 5.0000000
## S-R-CP4 25.0000000 10.000000 75.0000000 1.0000000
## S-A-TA4  2.1428571  0.000000  0.0000000 3.2857144
## B-A-MU1  1.6666666  0.000000  1.0000000 30.0000000
## B-A-GU1  0.0000000  0.000000  0.0000000 0.0000000
## B-R-PC2  4.0000000  1.000000  0.4000000 0.4000000
## B-A-MU2  2.4000001  0.000000  0.0000000 10.0000000
## B-A-GU2  0.0000000  0.000000  1.0000000 0.0000000
## B-R-PC3  3.3333333  0.000000  0.0000000 0.4444444
## B-A-MU3  0.0000000  0.000000  0.3333333 11.6666665
## B-A-GU3  5.5555556  0.000000 25.0000000 0.0000000
## B-R-PC4  2.0000000  1.250000  0.0000000 2.5000000
## B-A-MU4  0.0000000  5.000000 30.0000000 10.4444447
## B-A-GU4  1.4166667  0.000000 43.3333333 3.7500000

```

Salvando a matriz final particionada de dados ambientais.

```

write.table(m_amb_part, "m_amb_part.csv",
            sep = ";", dec = ".", #"\t",
            row.names = TRUE,
            quote = TRUE,
            append = FALSE)
m_amb_part <- read.csv("m_amb_part.csv",
                      sep = ";", dec = ".",
                      row.names = 1,
                      header = TRUE,
                      na.strings = NA)

```

13.8 Tabela de dados ambientais

Nessa etapa precisaremos da tabela de grupos. Selecionamos a aba **ambientalp**, porque nela constam os agrupamentos *a-priori* para os dados ambientais.

```

library(openxlsx)
t_grps <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizess/ppbio06-grup
                      rowNames = T, colNames = T,
                      sheet = "ambientalp")
str(t_grps)
#View(ppbio)
t_grps[1:4,1:4] #[1:4,1:4] mostra apenas as linhas e colunas de 1 a 4.

## 'data.frame':   23 obs. of  4 variables:
## $ area      : chr  "Serido" "Serido" "Serido" ...
## $ ambiente: chr  "rio"    "rio"    "acude"  "rio" ...
## $ UA        : chr  "CT"     "CP"    "TA"     "CT" ...
## $ coleta   : num  1 1 1 2 2 2 3 3 3 4 ...
##           area ambiente UA coleta
## S-R-CT1 Serido      rio CT      1
## S-R-CP1 Serido      rio CP      1
## S-A-TA1 Serido      acude TA    1
## S-R-CT2 Serido      rio CT      2

library(dplyr)
library(tidyr)
m_trab <- m_amb_part %>%
  rename_with(~ gsub("_", ".", .)) #apenas troquei o underscore pelo ponto.

m <- m_trab %>%
  group_by(Area = t_grps$area,
           Habitat = t_grps$ambiente,
           Ponto = t_grps$UA) %>%
  summarise(across(where(is.numeric),
                  list(mean = mean, min = min, max = max)),
             .groups = 'drop') %>%
  pivot_longer(
    cols = -c(Area, Habitat, Ponto),
    names_to = c("Variable", ".value"),
    names_sep = "_"
  )

m <- as.data.frame(m)
m_wide <- m %>%
  mutate(stat_string = ifelse(Variable == c("fq.w.vel"),
                             paste0(round(mean, 3), "(", round(min, 3), "-",
                             round(max, 3)), paste0(round(mean, 1), "(", round(min, 1), "-",
                             round(max, 1)),
                             unite("Location", Area, Habitat, Ponto, sep = "_") %>%
                             dplyr::select(Variable, Location, stat_string) %>% # find("select"), dplyr::select e

```

```

pivot_wider(names_from = Location, values_from = stat_string)

m_wide
m_wide <- as.data.frame(m_wide)
m_wide

#Exportando dados para Excel---
library(openxlsx)
write.xlsx(m_wide, file = "tabela de habitat.xlsx", rowNames = FALSE)
wb <- loadWorkbook("tabela de habitat.xlsx")
writeData(wb, sheet = "Sheet 1", x = m_wide)
saveWorkbook(wb, "tabela de habitat.xlsx", overwrite = TRUE)

#Escolher sumário de uma variável---
m
var <- "h.roots"
m[m$Variable == var, "mean"] #cada valor de var
summary(m[m$Variable == var, "mean"]) #sumário dos valores de var

#Escolher sumário de um grupo de variáveis do df m---
vars <- unique(grep("^h\\\\. ", m$Variable, value = TRUE))
summaries <- list() #criam uma lista vazia para guardar os sumários
# Loop para cada variável do grupo e guarda em summaries
for (var in vars) {
  summaries[[var]] <- summary(m[m$Variable == var, "mean"])
}
# var is a temporary variable used in the for loop to iterate through
# each variable name that starts with "h."

summaries
summary_table <- do.call(rbind, lapply(summaries, as.data.frame.list))
round(summary_table, 2)
#sink(file = "summary_h.txt", split = TRUE)
round(summary_table[order(summary_table$Mean, decreasing = FALSE), ], 2)
#sink()

#Tabela limpa
summary_table <- cbind(Variable = rownames(summary_table), summary_table)
rownames(summary_table) <- NULL
colnames(summary_table) <- c("Variable", "Min", "Q1", "Median", "Mean", "Q3", "Max")
summary_table

```

Apêndices

Sites consultados

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executa-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

Referências

Chapter 14

R Modulo 5 - Relativizações e transformações

RESUMO

Transformações e relativizações de dados são técnicas essenciais na análise de dados em Ecologia Numérica, permitindo comparar diferentes variáveis que estão em escalas diferentes ou que possuem unidades distintas.

Apresentação

O R é uma ferramenta poderosa para realização dessas operações, com diversos pacotes disponíveis para essa finalidade. Neste tutorial, iremos explorar algumas das principais funções do pacote `vegan`, que permite realizar transformações e relativizações de dados de forma simples e eficiente. Além disso, também utilizaremos outras bibliotecas importantes para reorganização e visualização de dados. O tutorial irá abordar desde as operações básicas, como filtragem e seleção de dados, até as transformações mais complexas, como a normalização de dados e a relativização por medidas de biomassa ou área. Com exemplos práticos e ilustrações gráficas, o objetivo é permitir que os usuários do R possam aplicar essas técnicas em suas próprias análises e estudos de Ecologia Numérica. Ao final do tutorial, espera-se que o usuário esteja apto a realizar transformações e relativizações de dados em suas próprias análises, aumentando a qualidade e a precisão dos resultados obtidos.

14.1 Organização básica

```
dev.off() #apaga os graficos, se houver algum
rm(list=ls(all=TRUE)) #limpa a memoria
cat("\014") #limpa o console
```

Instalando os pacotes necessários para esse módulo

```
install.packages("openxlsx") #importa arquivos do excel
install.packages("moments") #calcula assimetria e curtose dos dados
install.packages("vegan") #estatisticas para ecologia de comunidades
```

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

Definindo o diretório de trabalho e installando os pacotes necessários:

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

14.1.1 Importando a planilha

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações.

- Ajuste a primeira linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.
- Ajuste o parâmetro `sheet = "Sheet1"` para refletir a aba correta do arquivo .xlsx a ser importado.

```
library(openxlsx)
#dir <- getwd() #criamos um vetor com o diretório de trabalho
#shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
m_bruta <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06c-p
                      rowNames = T, colNames = T,
                      sheet = "Sheet1")
str(m_bruta)
#View(m_bruta)
m_bruta[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.

## 'data.frame': 26 obs. of 35 variables:
## $ ap-davis : num 0 0 0 0 0 0 0 0 0 ...
```

```

## $ as-bimac : num  1 99 194 19 23 142 5 46 206 16 ...
## $ as-fasci : num  0 0 55 0 1 3 1 0 64 0 ...
## $ ch-bimac : num  0 0 0 0 13 3 0 178 0 0 ...
## $ ci-ocela : num  0 0 0 0 0 0 40 0 0 13 ...
## $ ci-orien : num  0 0 5 0 0 69 9 0 25 24 ...
## $ co-macro : num  0 0 0 0 0 0 0 0 0 0 ...
## $ co-heter : num  0 0 1 0 0 0 0 0 0 0 ...
## $ cr-menez : num  0 0 14 0 0 4 0 0 8 0 ...
## $ cu-lepid : num  0 0 0 0 0 0 0 0 0 0 ...
## $ cy-gilbe : num  0 0 0 0 0 0 0 0 0 0 ...
## $ ge-brasi : num  0 0 3 0 0 0 0 0 1 0 ...
## $ he-margi : num  0 0 0 0 0 1 0 0 0 0 ...
## $ ho-malab : num  0 0 1 5 0 17 10 2 31 4 ...
## $ hy-pusar : num  0 0 9 2 0 43 2 0 11 0 ...
## $ le-melan : num  0 0 0 0 0 0 0 0 0 0 ...
## $ le-piau : num  0 0 3 0 0 1 3 0 2 1 ...
## $ le-taeni : num  0 0 0 0 0 0 0 0 0 0 ...
## $ mo-costa : num  0 0 0 0 0 0 0 0 0 0 ...
## $ mo-lepid : num  0 1 39 0 0 1 0 0 0 0 ...
## $ or-nilot : num  0 2 36 0 0 77 0 0 138 0 ...
## $ pa-manag : num  0 0 0 0 0 0 0 0 0 0 ...
## $ pimel-sp : num  0 0 6 0 0 0 0 0 0 0 ...
## $ po-retic : num  0 0 0 0 0 20 0 0 5 0 ...
## $ po-vivip : num  0 0 47 15 0 221 32 0 326 10 ...
## $ pr-brevi : num  9 0 5 0 1 15 5 2 164 0 ...
## $ ps-rhomb : num  0 0 0 0 0 0 0 0 1 0 ...
## $ ps-genise: num  0 0 0 0 0 0 0 0 1 0 ...
## $ se-heter : num  0 0 40 14 4 60 0 0 38 0 ...
## $ se-piaba : num  0 0 68 0 0 0 0 0 0 0 ...
## $ se-spilo : num  0 0 0 0 0 0 0 0 1 0 ...
## $ st-noton : num  0 0 1 0 0 25 0 0 115 0 ...
## $ sy-marmo : num  0 0 0 0 0 1 0 0 0 ...
## $ te-chalc : num  0 0 0 0 0 0 0 0 0 0 ...
## $ tr-signa : num  0 0 18 0 0 15 0 0 7 0 ...
##           ap-davis as-bimac as-fasci ch-bimac ci-ocela
## S-A-ZA1      0        1        0        0        0
## S-R-CC1      0       99        0        0        0
## S-R-CT1      0      194       55        0        0
## S-R-CP1      0       19        0        0        0
## S-A-TA1      0       23        1       13        0

```

Exibindo os dados importados (esses comando são “case-sensitive” `ignore.case(object)`).

```

#View(m_bruta)
print(m_bruta[1:8,1:8])
m_bruta[1:10,1:10]

```

```
str(m_bruta)
mode(m_bruta)
class(m_bruta)
```

Podemos exibir a planilha depois de ter sido importada para o ambiente R/RStudio usando as funções `View()`, `print()` ou `head()`. Note que essas funções são case-sensitive. A função `ignore.case()` é uma função do pacote `stringr` que modifica um padrão para que ele não considere o caso das letras nas correspondências. Por exemplo, se você quiser encontrar todas as ocorrências da letra “a” em um vetor de caracteres, independente de ser “A” ou “a”, você pode usar essa função.

```
#View(m_bruta)
print(m_bruta)
head(m_bruta)
```

A função `head()` no RStudio é uma forma de ver as primeiras ($n=6$) linhas de um objeto, como um vetor, uma matriz, um data frame ou uma lista. Ela é útil para ter uma ideia do conteúdo e da estrutura do objeto.

Também podemos explorar as características da planilha usando as funções `str()`, `mode()`, `class()` e `length()`. O número de observações ou tamanho do vetor depende do tipo de dados, se eles são uma matrix ou um data.frame.

Abreviações

No interesse de sistematizar o código R das várias matrizes que são comumente usadas em uma AMD, a tabela 14.1, a seguir, resume seus tipos e abreviações.

```
## Warning in table_info$align_vector[column] <-
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para
## para substituir não é um múltiplo do comprimento do substituto
```

14.2 Tamanho da matriz

Podemos agora calcular o número e a proporção de zeros na matriz usando as funções `sum()` e `length()` (Você pode pesquisar o que faz a função `length()` usando o comando `?length`).

```
range(m_bruta) #menor e maior valores
length(m_bruta) #no. de colunas
ncol(m_bruta) #no. de N colunas
nrow(m_bruta) #no. de M linhas
sum(lengths(m_bruta)) #soma os nos. de colunas
length(as.matrix(m_bruta)) #tamanho da matriz m x n
```

Table 14.1: Nomenclatura das matrizes em AMD em relação aos atributos das colunas

Nome	Atributos (colunas)	Abreviação no R
Matriz comunitaria	Os atributos são táxons ou OTU's (Unidades Taxonômicas Operacionais) (ex. espécies, gêneros, morfotipos)	m_com
Matriz ambiental	Os atributos são dados ambientais e variáveis físicas e químicas (ex. pH, condutividade, temperatura)	m_amb
Matriz de habitat	Os atributos são elementos da estrutura do habitat (ex. macróficas, algas, pedras, lama, etc)	m_hab
Matriz bruta	Os atributos ainda não receberam nenhum tipo de tratamento estatístico (valores brutos, como coletados)	m_bruta
Matriz transposta	Os atributos foram transpostos para as linhas	m_t
Matriz relativizada	Os atributos foram relativizados por um critério de tamanho ou de variação (ex. dividir os valores de cada coluna pela soma)	m_rel, m_relcol, m_rellin
Matriz transformada	Foi aplicado um operador matemático a todos os atributos (ex. raiz quadrada, log)	m_trns, m_log10, m_asrq
Matriz de distâncias	Matriz de m x m similaridades ou de distâncias (ex. Euclidiana, Manhattan, Bray-Curtis, etc)	m_dists, m_euclid, m_bray
Matriz de trabalho	Qualquer matriz que seja o foco da análise atual (ex. comunitária, relativizada, etc)	m_trab
Matriz particionada	Foram removidas linhas ou colunas (ex. linhas que são outliers e espécies zeradas)	m_part
Base de dados	Arquivo do Excel planilhado a partir de dados de campo ou de laboratório. Será manejada e particionada no R, para criar a Matriz bruta	ppbio06.xlsx, zoore- bio.xlsx, ben- tos06.xlsx

290 CHAPTER 14. R MODULO 5 - RELATIVIZAÇÕES E TRANSFORMAÇÕES

```
sum(m_bruta == 0) # número de observações igual a zero
sum(m_bruta > 0) # número de observações maiores que zero
zeros <- (sum(m_bruta == 0)/length(as.matrix(m_bruta)))*100 # proporção de zeros na matriz
zeros

## [1] 0 511
## [1] 35
## [1] 35
## [1] 26
## [1] 910
## [1] 910
## [1] 716
## [1] 194
## [1] 78.68132
```

Tabela que resume as informações geradas (14.2).

```
tamanho <- data.frame(
  Comando = c("range", "lenght", "m cols", "n linhas", "Tamanho", "Tamanho",
             "Zeros", "Nao zeros", "% Zeros"),
  Resultado = c(paste(range(m_bruta), collapse = " - "), length(m_bruta), ncol(m_bruta),
                nrow(m_bruta), sum(lengths(m_bruta)), length(as.matrix(m_bruta)), sum(m_bruta == 0),
                sum(m_bruta > 0), round(zeros, 1)))
tamanho

##      Comando Resultado
## 1      range    0 - 511
## 2     lenght      35
## 3      m cols      35
## 4     n linhas      26
## 5      Tamanho     910
## 6      Tamanho     910
## 7      Zeros       716
## 8    Nao zeros      194
## 9      % Zeros    78.7
```

Ou seja, temos uma matriz de tamanho $m \times n$ igual a 35 atributos por 26 objetos, onde 78.68% dos valores da matriz são iguais a zero!

14.3 Reset point 1

```
m_trab <- m_bruta
```

Aqui cria-se um novo objeto do R (`m_trab`, ou a matriz de trabalho, para esse

Table 14.2: Resumo das informações geradas.

Comando	Resultado
range	0 - 511
length	35
m_cols	35
n_linhas	26
Tamanho	910
Tamanho	910
Zeros	716
Nao zeros	194
% Zeros	78.7

momento) que substitui a matriz de dados original, por uma nova matriz que pode ser a matriz relativizada, transformada, transposta, etc. Dessa forma, mantemos a matriz de dados original caso precisemos dela novamente (Veja a Tabela Tabela @ref(tab:5tblm_2)).

14.4 Cálculo da matriz de distâncias

Agora vamos calcular a matriz de distâncias euclidiana (LEGENDRE; LEGENDRE (1998), BORCARD; GILLET; LEGENDRE (2018)) entre as UA's usando a função `dist()` (pesquise o que faz essa função usando o comando `?dist`).

```
m_dists <- dist(m_trab, method = "euclidean", diag = TRUE, upper = FALSE)
```

Pronto, calculamos a matriz de distâncias (euclidiana). Agora podemos visualizar a matriz.

```
#m_dists
str(m_dists)
mode(m_dists)
class(m_dists)
length(as.matrix(m_dists))
as.matrix(m_dists)[1:10, 1:10]

##  'dist' num [1:325] 98.4 228.5 29.2 27.1 294.1 ...
##  - attr(*, "Size")= int 26
##  - attr(*, "Labels")= chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
##  - attr(*, "Diag")= logi TRUE
##  - attr(*, "Upper")= logi FALSE
##  - attr(*, "method")= chr "euclidean"
##  - attr(*, "call")= language dist(x = m_trab, method = "euclidean", diag = TRUE, upper = FALSE)
## [1] "numeric"
```

292CHAPTER 14. R MODULO 5 - RELATIVIZAÇÕES E TRANSFORMAÇÕES

```

## [1] "dist"
## [1] 676
##          S-A-ZA1   S-R-CC1   S-R-CT1   S-R-CP1   S-A-TA1   S-R-CT2   S-R-CP2
## S-A-ZA1    0.00000  98.43780 228.5235  29.24038  27.09243 294.0629  53.40412
## S-R-CC1   98.43780   0.00000 154.2433  82.79493  77.25283 261.3905 108.10180
## S-R-CT1  228.52352 154.24331   0.0000 208.52338 209.69263 224.9133 224.09150
## S-R-CP1   29.24038  82.79493 208.5234   0.00000 23.25941 271.4922 49.22398
## S-A-TA1   27.09243  77.25283 209.6926  23.25941   0.00000 283.7869 57.82733
## S-R-CT2  294.06292 261.39051 224.9133 271.49217 283.78689   0.0000 268.95167
## S-R-CP2   53.40412 108.10180 224.0915  49.22398  57.82733 268.9517  0.00000
## S-A-TA2  183.74439 185.75791 261.8301 181.24845 166.66133 326.3327 190.15257
## S-R-CT3  460.42155 428.64437 366.6429 443.90877 453.59674 238.3799 438.54532
## S-R-CP3   34.17601  88.06816 215.2557 31.32092 33.13608 275.0436 40.37326
##          S-A-TA2   S-R-CT3   S-R-CP3
## S-A-ZA1  183.7444 460.4215 34.17601
## S-R-CC1  185.7579 428.6444 88.06816
## S-R-CT1  261.8301 366.6429 215.25566
## S-R-CP1  181.2484 443.9088 31.32092
## S-A-TA1  166.6613 453.5967 33.13608
## S-R-CT2  326.3327 238.3799 275.04363
## S-R-CP2  190.1526 438.5453 40.37326
## S-A-TA2   0.0000 478.9008 182.86060
## S-R-CT3  478.9008   0.0000 449.24826
## S-R-CP3  182.8606 449.2483   0.00000

```

Uma matriz do tipo `dist` no R é um objeto que armazena as distâncias entre as linhas de uma matriz ou um data frame. Ela é criada pela função `dist()`, que calcula as distâncias usando diferentes medidas, como “euclidean”, “manhattan”, “canberra”, “binary” ou “minkowski” (HORTON; KLEINMAN, 2015).

```

range(m_trab)
range(m_dists)
min(m_dists)
max(m_dists)
mean(m_dists) #CENTROIDE!! ou Grand mean
sd(m_dists) #standard deviation
centroide <- mean(m_dists)
centroide

## [1] 0 511
## [1] 16.27882 707.32666
## [1] 16.27882
## [1] 707.3267
## [1] 285.9043
## [1] 172.9708
## [1] 285.9043

```

A função `mean()` calcula a média de todos os valores da matriz de distâncias, ou seja, a média multivariada, que é o centroíde. Nesse caso o **centroide assume o valor de 285.9**. Usamos agora a fórmula $m*(m-1)/2$, onde m é o no. de objetos sendo comparados, para calcular quantas distâncias temos na nossa matriz.

```
length(m_dists)
m <- nrow(as.matrix(m_dists))
m
m*(m-1)/2
summary(m_dists)

## [1] 325
## [1] 26
## [1] 325
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
## 16.28 142.70 242.82 285.90 423.64 707.33
```

Temos então que m é 26 objetos (ou linhas), e portanto, a matriz de distâncias tem 325 valores. Fazemos agora um breve sumário do que foi calculado até agora com base na matriz de distâncias.

```
Sumario1 <- cbind(min(m_dists),
                    max(m_dists),
                    sd(m_dists),
                    mean(m_dists),
                    length(m_dists))
colnames(Sumario1) <- c("Minimo", "Maximo", "Desv.Padr", "Media", "m(m-1)/2")
rownames(Sumario1) <- ("Valores")
Sumario1

##           Minimo   Maximo Desv.Padr     Media m(m-1)/2
## Valores 16.27882 707.3267 172.9708 285.9043      325
```

A matriz `m_dists` está como uma classe `dist`, mas podemos transformá-la em uma classe `matrix` usando a função `as.matrix()`.

```
dist <- (as.matrix(m_dists))
#View(dist)
str(dist)
mode(dist)
class(dist)
#head(dist)
dist[1:5, 1:5] #mostra as 5 primeiras linhas e colunas da matriz

## num [1:26, 1:26] 0 98.4 228.5 29.2 27.1 ...
## - attr(*, "dimnames")=List of 2
```

```

##   . . $ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
##   . . $ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
## [1] "numeric"
## [1] "matrix" "array"
##           S-A-ZA1   S-R-CC1   S-R-CT1   S-R-CP1   S-A-TA1
## S-A-ZA1    0.00000  98.43780 228.5235  29.24038 27.09243
## S-R-CC1   98.43780   0.00000 154.2433  82.79493 77.25283
## S-R-CT1  228.52352 154.24331   0.00000 208.52338 209.69263
## S-R-CP1  29.24038  82.79493 208.5234   0.00000 23.25941
## S-A-TA1  27.09243  77.25283 209.6926  23.25941   0.00000

```

Visualizando a matriz de distâncias, observamos que ela é uma matriz quadrada que contém as distâncias entre cada par de elementos do conjunto de dados. A matriz de distâncias terá dimensão $m \times m$, e cada elemento da matriz será a distância entre cada par de observações ou objetos. A matriz de distâncias é simétrica, pois a distância entre i e j é igual à distância entre j e i . A matriz de distâncias também tem diagonal zero, pois a distância entre uma observação e ela mesma é zero.

14.5 Distribuição de frequência da matriz de distâncias

Por fim, vamos produzir uma distribuição de frequência da matriz de distâncias, mas antes vamos explorar alguns parâmetros básicos da matriz de distâncias, como o intervalo, a média e o desvio padrão. Para isso, usamos as funções `range()`, `mean()` e `sd()` do R.

14.5.1 Calculando alguns parâmetros básicos

```

range(m_dists) # intervalo dos valores da matriz
mean(m_dists) # média dos valores da matriz
sd(m_dists) # desvio padrão dos valores da matriz

```

```

## [1] 16.27882 707.32666
## [1] 285.9043
## [1] 172.9708

```

Esses comandos mostram que o menor, maior valores na matriz de distância foram 16.28, 707.33, enquanto que a média foi de 285.9 com um desvio padrão de 172.97.

IMPORTANTE Lembre que no R e RStudio, a vírgula é usada como separador de decimal e o ponto é apenas indicador de milhar. Por exemplo, 2.500,50 lê-se dois mil e quinhentos e 50 décimos.

Também podemos calcular a assimetria e a curtose da matriz de distâncias, que são medidas que indicam o grau de desvio da normalidade da distribuição. Para isso, precisamos carregar o pacote **moments**, que contém as funções **skewness()** (assimetria) e **kurtosis()** (curtose).

14.5.2 Calculando a assimetria e a curtose

```
library(moments)
sk <- skewness(as.matrix(m_dists)) #calcula a assimetria da matriz
ku <- kurtosis(as.matrix(m_dists)) #calcula a curtose da matriz
sku <- cbind(sk,ku) # junta os dois valores em um vetor
colnames(sku) <- c("assimetria", "curtose") # nomeia as colunas do vetor
sku[1:10, ] # mostra as primeiras 10 linhas desse vetor
summary(sku) # mostra um resumo estatístico do vetor

##           assimetria    curtose
## S-A-ZA1   1.03617037  3.101029
## S-R-CC1   1.21511849  3.852134
## S-R-CT1   0.91278496  6.263889
## S-R-CP1   1.06839482  3.177206
## S-A-TA1   1.07960546  3.220821
## S-R-CT2   0.05290597  6.265832
## S-R-CP2   1.08571040  3.241712
## S-A-TA2   1.25111833  4.838956
## S-R-CT3   -2.35918827 10.736848
## S-R-CP3   1.06511174  3.178170
##           assimetria    curtose
##  Min.   : -3.9660  Min.   : 2.946
##  1st Qu.:  0.1891  1st Qu.: 3.226
##  Median :  1.0531  Median : 3.793
##  Mean   :  0.3625  Mean   : 5.248
##  3rd Qu.:  1.1593  3rd Qu.: 5.659
##  Max.   :  1.2826  Max.   :18.799
```

14.6 Distribuição de frequências da matriz de distâncias

Agora que temos uma ideia dos parâmetros básicos da matriz de distâncias, vamos entender sua distribuição de frequência usando um histograma e um boxplot. O histograma mostra a frequência relativa de cada intervalo de valores da matriz, enquanto o boxplot mostra a mediana, os quartis e os valores extremos da matriz.

Para fazer os gráficos, usamos as funções **hist()** e **boxplot.default()** do R.

296 CHAPTER 14. R MODULO 5 - RELATIVIZAÇÕES E TRANSFORMAÇÕES

Também usamos a função `curve()` para sobrepor uma curva de normalidade ao histograma, e a função `par` para ajustar os parâmetros gráficos.

Nas linhas `xlim = ()`, # limites dos eixos do código abaixo, os limites dos eixos do gráfico a ser criado foram definidos entre 0 e 707.3266572. No seu caso, você deve ajustar os valores de `xlim` para que eles sejam os limites dos valores da sua matriz, definidos pela linha `range(dist)` (*primeira linha do código abaixo*).

Em resumo, o código abaixo plota um histograma e um boxplot da distribuição de dados armazenada na matriz de distâncias. Também é adicionada uma curva normal teórica ao histograma utilizando a média e o desvio padrão dos dados (Figura 14.1). As funções `floor(min(m_dists))` e `ceiling(max(m_dists))` definem os limites dos graficos pelo valores mínimo e máximo do objeto.

```
range(m_dists)
par(mfrow=c(2,1)) # esse comando faz os gráficos aparecerem um acima do outro
hist(m_dists,
      breaks = 20, # determina o número de colunas do histograma
      xlim = range(floor(min(m_dists)), ceiling(max(m_dists))), # limites dos eixos
      xlab = "Distr. de Frequências",
      freq = FALSE, # mostra as frequências relativas em vez das absolutas
      main = "Histograma e boxplot da matriz de distâncias")
curve(dnorm(x, mean=mean(m_dists), sd=sd(m_dists)), add=TRUE) # sobrepõe a curva de no
boxplot.default(m_dists, horizontal = TRUE, frame = FALSE,
                 xlab="Distr. de Frequências",
                 ylim=c(floor(min(m_dists)), ceiling(max(m_dists)))) # limites dos eixo
```

```
## [1] 16.27882 707.32666
```

IMPORTANTE [Atenção:] Se você recebeu a mensagem de erro “Error in plot.new() : figure margins too large”, aumente o tamanho da janela do gráfico e execute o bloco de comandos novamente.

O comando abaixo apaga os gráficos.

```
dev.off()
```

14.7 Relativizando e transformando a base de dados

Na sequencia de códigos a seguir o R realiza relativizações e transformações em dados biológicos (SOKAL; ROHLF (1995)). Vamos ver passo a passo o que cada seção do código faz. Os códigos para as demais principais relativizações estão AQUI

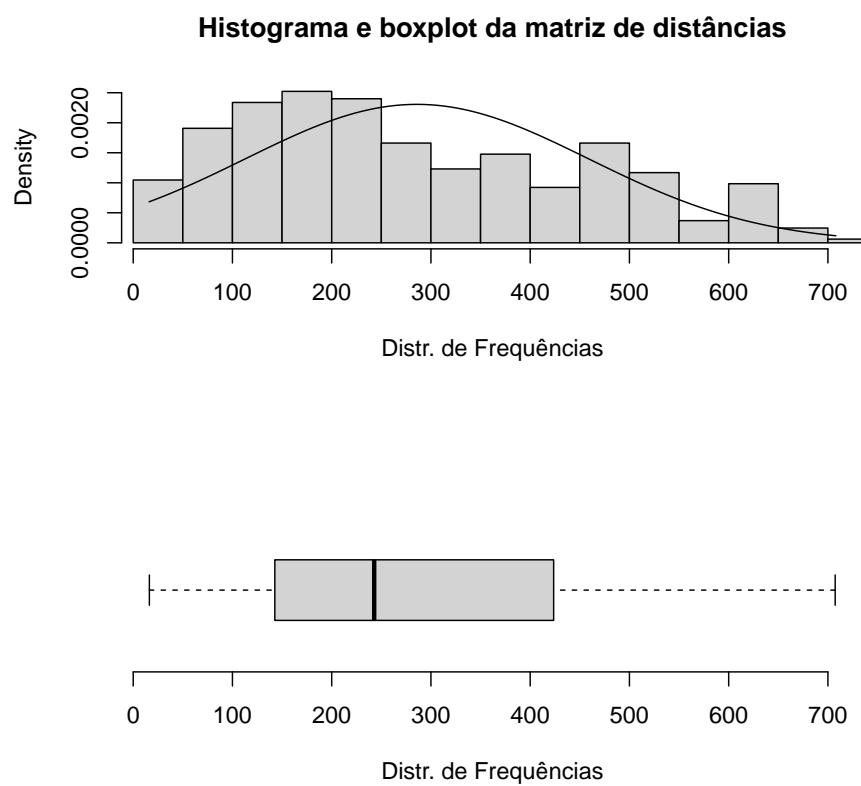


Figure 14.1: Distribuições de frequências da matriz de distâncias

```
library(vegan)
```

Essa linha carrega o pacote `vegan` no ambiente R, permitindo o uso de suas funções. O pacote `vegan` é um pacote do R para análise de ecologia numérica e comunidades biológicas. Ele fornece uma ampla variedade de funções para análise de dados ecológicos, como análise de diversidade, ordenação de espécies, análise de similaridade, entre outras.

A função `decostand()` é usada para relativizar os dados armazenados na matriz de trabalho (`m_trab`). O parâmetro `method` especifica o método de relativização, neste caso “*total*” (relativização pelo total de cada coluna). Outros métodos disponíveis são *max*, *normalize*, *range* e *rankm* (@ref(tab:(5tbl-rel))). O parâmetro `MARGIN` especifica em qual dimensão os cálculos devem ser feitos (*1 para linhas, 2 para colunas*). Neste caso, `MARGIN=2` (colunas). A variável `m_relcol` armazena o resultado da relativização. Sokal e Rohlf (SOKAL; ROHLF (1987)) tratam dos diversos tipos de relativizações e transformações em dados ecológicos.

14.7.1 Relativização pelo total

```
m_relcol <- decostand(m_trab,
                       method="total",
                       MARGIN = 2)
#View(m_relcol)
m_relcol

##          ap-davis    as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien
## S-A-ZA1 0.0000000 0.0004191115 0.0000000000 0.0000000000 0.000000000 0.000000000
## S-R-CC1 0.0000000 0.0414920369 0.0000000000 0.0000000000 0.000000000 0.000000000
## S-R-CT1 0.0000000 0.0813076278 0.348101266 0.0000000000 0.000000000 0.03496503
## S-R-CP1 0.0000000 0.0079631182 0.0000000000 0.0000000000 0.000000000 0.000000000
## S-A-TA1 0.0000000 0.0096395641 0.006329114 0.018439716 0.000000000 0.000000000
## S-R-CT2 0.0000000 0.0595138307 0.018987342 0.004255319 0.000000000 0.48251748
## S-R-CP2 0.0000000 0.0020955574 0.006329114 0.000000000 0.57142857 0.06293706
## S-A-TA2 0.0000000 0.0192791282 0.000000000 0.252482270 0.000000000 0.000000000
## S-R-CT3 0.0000000 0.0863369656 0.405063291 0.000000000 0.000000000 0.17482517
## S-R-CP3 0.0000000 0.0067057837 0.000000000 0.000000000 0.18571429 0.16783217
## S-A-TA3 0.0000000 0.0980720872 0.044303797 0.337588652 0.000000000 0.000000000
## S-R-CT4 0.0000000 0.0000000000 0.006329114 0.000000000 0.000000000 0.03496503
## S-R-CP4 0.0000000 0.0000000000 0.000000000 0.000000000 0.15714286 0.04195804
## S-A-TA4 0.0000000 0.1651299246 0.000000000 0.387234043 0.000000000 0.000000000
## B-A-MU1 0.0000000 0.0050293378 0.000000000 0.000000000 0.000000000 0.000000000
## B-R-ET1 0.0000000 0.0012573345 0.000000000 0.000000000 0.000000000 0.000000000
## B-A-GU1 0.0000000 0.0008382230 0.012658228 0.000000000 0.000000000 0.000000000
```

```

## B-R-PC2 0.1851852 0.0184409053 0.000000000 0.000000000 0.02857143 0.00000000
## B-A-MU2 0.0000000 0.0414920369 0.000000000 0.000000000 0.00000000 0.00000000
## B-A-GU2 0.0000000 0.000000000 0.000000000 0.000000000 0.00000000 0.00000000
## B-R-PC3 0.8148148 0.0314333613 0.044303797 0.000000000 0.05714286 0.00000000
## B-A-MU3 0.0000000 0.2141659681 0.000000000 0.000000000 0.00000000 0.00000000
## B-A-GU3 0.0000000 0.0025146689 0.000000000 0.000000000 0.00000000 0.00000000
## B-R-PC4 0.0000000 0.0029337804 0.107594937 0.000000000 0.00000000 0.00000000
## B-A-MU4 0.0000000 0.0984911987 0.000000000 0.000000000 0.00000000 0.00000000
## B-A-GU4 0.0000000 0.0054484493 0.000000000 0.000000000 0.00000000 0.00000000
## co-macro co-heter cr-menez cu-lepid cy-gilbe ge-brasi he-margi
## S-A-ZA1      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-R-CC1      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-R-CT1      0      1 0.50000000 0 0.0000000 0.0029411765 0.0
## S-R-CP1      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-A-TA1      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-R-CT2      0      0 0.14285714 0 0.0000000 0.000000000000 0.5
## S-R-CP2      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-A-TA2      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-R-CT3      0      0 0.28571429 0 0.0000000 0.0009803922 0.0
## S-R-CP3      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-A-TA3      1      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-R-CT4      0      0 0.03571429 0 0.3816794 0.0029411765 0.5
## S-R-CP4      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## S-A-TA4      0      0 0.03571429 0 0.0000000 0.0009803922 0.0
## B-A-MU1      0      0 0.00000000 0 0.0000000 0.1862745098 0.0
## B-R-ET1      0      0 0.00000000 0 0.0000000 0.000000000000 0.0
## B-A-GU1      0      0 0.00000000 0 0.0000000 0.0068627451 0.0
## B-R-PC2      0      0 0.00000000 0 0.0000000 0.0078431373 0.0
## B-A-MU2      0      0 0.00000000 0 0.0000000 0.0656862745 0.0
## B-A-GU2      0      0 0.00000000 0 0.0000000 0.0225490196 0.0
## B-R-PC3      0      0 0.00000000 1 0.0000000 0.0156862745 0.0
## B-A-MU3      0      0 0.00000000 0 0.0000000 0.1421568627 0.0
## B-A-GU3      0      0 0.00000000 0 0.0000000 0.0313725490 0.0
## B-R-PC4      0      0 0.00000000 0 0.6183206 0.0049019608 0.0
## B-A-MU4      0      0 0.00000000 0 0.0000000 0.4990196078 0.0
## B-A-GU4      0      0 0.00000000 0 0.0000000 0.0098039216 0.0
## ho-malab    hy-pusar le-melan   le-piau le-taeni mo-costa mo-lepid
## S-A-ZA1 0.000000000 0.00000000 0 0.00000000 0 0 0.00000000
## S-R-CC1 0.000000000 0.00000000 0 0.00000000 0 0 0.02439024
## S-R-CT1 0.009174312 0.12676056 0 0.20000000 0 0 0.95121951
## S-R-CP1 0.045871560 0.02816901 0 0.00000000 0 0 0.00000000
## S-A-TA1 0.000000000 0.00000000 0 0.00000000 0 0 0.00000000
## S-R-CT2 0.155963303 0.60563380 0 0.06666667 0 0 0.02439024
## S-R-CP2 0.091743119 0.02816901 0 0.20000000 0 0 0.00000000
## S-A-TA2 0.018348624 0.00000000 0 0.00000000 0 0 0.00000000
## S-R-CT3 0.284403670 0.15492958 0 0.13333333 0 0 0.00000000

```

300CHAPTER 14. R MODULO 5 - RELATIVIZAÇÕES E TRANSFORMAÇÕES

```

## S-R-CP3 0.036697248 0.00000000          0 0.06666667    0      0 0.00000000
## S-A-TA3 0.183486239 0.00000000          0 0.00000000    0      0 0.00000000
## S-R-CT4 0.036697248 0.04225352          0 0.00000000    0      0 0.00000000
## S-R-CP4 0.018348624 0.00000000          0 0.13333333    0      0 0.00000000
## S-A-TA4 0.082568807 0.00000000          0 0.13333333    0      0 0.00000000
## B-A-MU1 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-R-ET1 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-A-GU1 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-R-PC2 0.000000000 0.00000000          1 0.00000000    1      0 0.00000000
## B-A-MU2 0.009174312 0.00000000          0 0.00000000    0      0 0.00000000
## B-A-GU2 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-R-PC3 0.018348624 0.01408451          0 0.00000000    0      1 0.00000000
## B-A-MU3 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-A-GU3 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-R-PC4 0.009174312 0.00000000          0 0.06666667    0      0 0.00000000
## B-A-MU4 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
## B-A-GU4 0.000000000 0.00000000          0 0.00000000    0      0 0.00000000
##           or-nilot   pa-manag pimel-sp   po-retic   po-vivip   pr-brevi
## S-A-ZA1 0.000000000 0.000000000          0 0.00000000 0.000000000 0.032258065
## S-R-CC1 0.002358491 0.000000000          0 0.00000000 0.000000000 0.000000000
## S-R-CT1 0.042452830 0.000000000          1 0.00000000 0.048057260 0.017921147
## S-R-CP1 0.000000000 0.000000000          0 0.00000000 0.015337423 0.000000000
## S-A-TA1 0.000000000 0.000000000          0 0.00000000 0.000000000 0.003584229
## S-R-CT2 0.090801887 0.000000000          0 0.05249344 0.225971370 0.053763441
## S-R-CP2 0.000000000 0.000000000          0 0.00000000 0.032719836 0.017921147
## S-A-TA2 0.000000000 0.000000000          0 0.00000000 0.000000000 0.007168459
## S-R-CT3 0.162735849 0.000000000          0 0.01312336 0.333333333 0.587813620
## S-R-CP3 0.000000000 0.000000000          0 0.00000000 0.010224949 0.000000000
## S-A-TA3 0.000000000 0.000000000          0 0.00000000 0.000000000 0.000000000
## S-R-CT4 0.086084906 0.000000000          0 0.00000000 0.028629857 0.211469534
## S-R-CP4 0.000000000 0.000000000          0 0.00000000 0.081799591 0.000000000
## S-A-TA4 0.001179245 0.000000000          0 0.00000000 0.000000000 0.010752688
## B-A-MU1 0.007075472 0.000000000          0 0.00000000 0.000000000 0.000000000
## B-R-ET1 0.009433962 0.001805054          0 0.08923885 0.000000000 0.000000000
## B-A-GU1 0.003537736 0.019855596          0 0.00000000 0.000000000 0.000000000
## B-R-PC2 0.005896226 0.000000000          0 0.00000000 0.000000000 0.032258065
## B-A-MU2 0.001179245 0.000000000          0 0.02624672 0.008179959 0.000000000
## B-A-GU2 0.042452830 0.184115523          0 0.00000000 0.000000000 0.000000000
## B-R-PC3 0.076650943 0.000000000          0 0.00000000 0.000000000 0.021505376
## B-A-MU3 0.012971698 0.000000000          0 0.12073491 0.049079755 0.003584229
## B-A-GU3 0.291273585 0.451263538          0 0.00000000 0.000000000 0.000000000
## B-R-PC4 0.010613208 0.000000000          0 0.00000000 0.000000000 0.000000000
## B-A-MU4 0.001179245 0.000000000          0 0.69816273 0.166666667 0.000000000
## B-A-GU4 0.152122642 0.342960289          0 0.00000000 0.000000000 0.000000000
##           ps-rhomb ps-genise se-heter se-piaba se-spilo st-noton sy-marmo
## S-A-ZA1          0          0 0.000000000          0          0 0.000000000          0

```

```

## S-R-CC1      0      0 0.00000000      0      0 0.000000000      0
## S-R-CT1      0      0 0.13513514      1      0 0.004878049      0
## S-R-CP1      0      0 0.04729730      0      0 0.000000000      0
## S-A-TA1      0      0 0.01351351      0      0 0.000000000      0
## S-R-CT2      0      0 0.20270270      0      0 0.121951220      0
## S-R-CP2      0      0 0.00000000      0      0 0.000000000      1
## S-A-TA2      0      0 0.00000000      0      0 0.000000000      0
## S-R-CT3      1      1 0.12837838      0      1 0.560975610      0
## S-R-CP3      0      0 0.00000000      0      0 0.000000000      0
## S-A-TA3      0      0 0.00000000      0      0 0.000000000      0
## S-R-CT4      0      0 0.01013514      0      0 0.312195122      0
## S-R-CP4      0      0 0.01013514      0      0 0.000000000      0
## S-A-TA4      0      0 0.00000000      0      0 0.000000000      0
## B-A-MU1      0      0 0.00000000      0      0 0.000000000      0
## B-R-ET1      0      0 0.00000000      0      0 0.000000000      0
## B-A-GU1      0      0 0.00000000      0      0 0.000000000      0
## B-R-PC2      0      0 0.03378378      0      0 0.000000000      0
## B-A-MU2      0      0 0.00000000      0      0 0.000000000      0
## B-A-GU2      0      0 0.00000000      0      0 0.000000000      0
## B-R-PC3      0      0 0.31418919      0      0 0.000000000      0
## B-A-MU3      0      0 0.00000000      0      0 0.000000000      0
## B-A-GU3      0      0 0.00000000      0      0 0.000000000      0
## B-R-PC4      0      0 0.10472973      0      0 0.000000000      0
## B-A-MU4      0      0 0.00000000      0      0 0.000000000      0
## B-A-GU4      0      0 0.00000000      0      0 0.000000000      0
##           te-chalc   tr-signa
## S-A-ZA1 0.0000000 0.00000000
## S-R-CC1 0.0000000 0.00000000
## S-R-CT1 0.0000000 0.08653846
## S-R-CP1 0.0000000 0.00000000
## S-A-TA1 0.0000000 0.00000000
## S-R-CT2 0.0000000 0.07211538
## S-R-CP2 0.0000000 0.00000000
## S-A-TA2 0.0000000 0.00000000
## S-R-CT3 0.0000000 0.03365385
## S-R-CP3 0.0000000 0.00000000
## S-A-TA3 0.0000000 0.00000000
## S-R-CT4 0.0000000 0.67788462
## S-R-CP4 0.0000000 0.00000000
## S-A-TA4 0.0000000 0.00000000
## B-A-MU1 0.0000000 0.00000000
## B-R-ET1 0.0000000 0.00000000
## B-A-GU1 0.0000000 0.00000000
## B-R-PC2 0.5671642 0.11057692
## B-A-MU2 0.0000000 0.00000000
## B-A-GU2 0.0000000 0.00000000

```

```

## B-R-PC3 0.4328358 0.0000000
## B-A-MU3 0.0000000 0.0000000
## B-A-GU3 0.0000000 0.0000000
## B-R-PC4 0.0000000 0.01923077
## B-A-MU4 0.0000000 0.0000000
## B-A-GU4 0.0000000 0.0000000

#colSums(m_relcol)
#range(m_relcol)

```

14.7.2 Transformação pelo arcoseno da raiz quadrada

A seguir é realizada a transformação pelo arcoseno da raiz quadrada, apropriada para ados em proporção como os ados previamente relativizados. Para cada transformação, a função é aplicada aos dados de interesse e o resultado é armazenado em uma nova variável. A função `View()` é usada para visualizar o resultado de cada transformação na forma de uma tabela. Os códigos para as demais principais transformações estão AQUI

```

m_relcol_asrq <- asin(sqrt(m_relcol))
#View(m_relcol_asrq)
m_relcol_asrqi <- 2/pi*(asin(sqrt(m_relcol)))
#View(m_relcol_asrqi)
#colSums(m_relcol_asrqi)
#range(m_relcol_asrqi)

```

14.8 Reset point 2

Aqui estabelecemos a matriz transformada depois de ter sido relativizada como `m_trns`.

```

m_trns <- m_relcol_asrqi
#m_dists <- dist(m_trns, method = "euclidian", diag = TRUE, upper = FALSE)

```

Agora é com você... Refaça toda a análise com as relativizações e transformações adequadas para a matriz de dados fornecida.

Table 14.3: Métodos de relativização da função ‘decostand’

Função	Descrição	Uso adequado
max	Divide cada valor pela maior observação da coluna em que está localizado. Todos os valores resultantes serão menores ou iguais a 1. Adequada para análise em escala relativa	Quando a escala de cada variável é conhecida e a análise deve ser feita em uma escala relativa
normalize	Divide cada valor pelo comprimento do vetor. Isso resulta em valores cujo comprimento é sempre igual a 1. Adequada para análise de dados de frequência	Para análise de dados de frequência
range	Subtrai o valor mínimo da coluna de cada valor e, em seguida, divide pelo intervalo (ou amplitude) dos valores da coluna. Isso resulta em valores entre 0 e 1.	Quando a amplitude dos valores em cada coluna é importante para a análise
rankm	Transforma os valores em suas posições dentro da coluna, em ordem crescente. O menor valor recebe o valor 1 e o maior valor recebe o valor igual ao comprimento da coluna.	Quando a escala absoluta dos valores não é importante, mas a ordem dos valores é significativa.

Apêndices

Relativizações

Total das colunas

```
m_relcol <- decostand(m_bruta,
                      method="total",
                      MARGIN = 2)
View(m_relcol)
m_relcol
```

A função `decostand()` é usada para relativizar os dados armazenados no vetor `m_bruta`. O parâmetro `method` especifica o método de relativização, neste caso “total” (relativização pelo total de cada coluna). Outros métodos disponíveis são `max`, `normalize`, `range` e `rankm` (Figura 14.3). O parâmetro `MARGIN` especifica em qual dimensão os cálculos devem ser feitos (1 para linhas, 2 para colunas). Neste caso, `MARGIN=2` (colunas). A variável `m_relcol` armazena o resultado da relativização. Sokal e Rohlf (1995) tratam dos diversos tipos de relativizações e transformações em dados ecológicos.

```
## Warning in table_info$align_vector[column] <-
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para
## para substituir não é um múltiplo do comprimento do substituto
```

Transformações

Essa seção mostra as diversas transformações nos dados armazenados no vetor `m_bruta`. Para cada transformação, a função é aplicada aos dados da matriz e o resultado é armazenado em uma nova variável. A função `View()` pode ser usada para visualizar o resultado de cada transformação na forma de uma tabela. Ela mostra o código que realiza diversas transformações na matriz de dados. Para cada transformação, a função é aplicada aos dados e o resultado é armazenado em uma nova variável.

Log10

Não aceita zeros, porque o log de zero é indeterminado. Por isso devemos substituir os valores com erro (infinito negativo) por zero, usando a função `replace()`.

```
m_lg10 <- log10(m_bruta)
#View(m_lg10)
m_lg10 <- replace(m_lg10, is.infinite(m_lg10), 0)
head(m_lg10)
```

Log(x+1)

Desaconselhável para valores que variam entre 0-1.

```
m_lgx1 <- log(m_bruta+1)
#View(m_lgx1)
```

Exponenciais

“Power transformations”, onde o valor de `p` estabelece a compressão dos dados.

```
m_bruta_P1 <- m_bruta^1 #p=1
#View(m_bruta_P1)
m_bruta_P05 <- m_bruta^0.5 #p=0,5
#View(m_bruta_P05)
m_bruta_P01 <- m_bruta^0.1 #p=0,1
#View(m_bruta_P01)
```

Presença/Ausência

```
m_pa <- (m_bruta>0)*1L
#View(m_pa)
```

Raiz quadrada

```
m_rq <- sqrt(m_bruta)
#View(m_rq)
```

Arcoseno da raiz quadrada

Os valores de entrada tem que variar entre 0 e 1. Ideal para matrizes relativizadas. Duas formulações estão disponíveis, com e sem a multiplicação por 2/pi. A segunda opção costuma ter melhor efeito na normalização dos dados.

```
m_asrq <- asin(sqrt(m_relcol))
#View(m_asrq)
m_asrqpi <- 2/pi*(asin(sqrt(m_relcol)))
View(m_asrqpi)
```

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executá-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

```
## dev.off() #apaga os graficos, se houver algum
## rm(list=ls(all=TRUE)) #limpa a memória
## cat("\014") #limpa o console
## install.packages("openxlsx") #importa arquivos do excel
## install.packages("moments") #calcula assimetria e curtose dos dados
## install.packages("vegan") #estatísticas para ecologia de comunidades
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
library(openxlsx)
#dir <- getwd() #criamos um vetor com o diretório de trabalho
#shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
m_bruta <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/peixes06.xlsx",
                      rowNames = T, colNames = T,
                      sheet = "Sheet1")
str(m_bruta)
#View(m_bruta)
m_bruta[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
## #View(m_bruta)
## print(m_bruta[1:8,1:8])
## m_bruta[1:10,1:10]
## str(m_bruta)
```

306 CHAPTER 14. R MODULO 5 - RELATIVIZAÇÕES E TRANSFORMAÇÕES

```

## mode(m_bruta)
## class(m_bruta)
#View(m_bruta)
print(m_bruta)
head(m_bruta)
range(m_bruta) #menor e maior valores
length(m_bruta) #no. de colunas
ncol(m_bruta) #no. de N colunas
nrow(m_bruta) #no. de M linhas
sum(lengths(m_bruta)) #soma os nos. de colunas
length(as.matrix(m_bruta)) #tamanho da matriz m x n
sum(m_bruta == 0) # número de observações igual a zero
sum(m_bruta > 0) # número de observações maiores que zero
zeros <- (sum(m_bruta == 0)/length(as.matrix(m_bruta)))*100 # proporção de zeros na matriz
zeros
tamanho <- data.frame(
  Função = c("range", "length", "m cols", "n linhas", "Tamanho", "Tamanho",
            "Zeros", "Nao zeros", "% Zeros"),
  Resultado = c(paste(range(m_bruta), collapse = " - "), length(m_bruta), ncol(m_bruta),
                nrow(m_bruta), sum(lengths(m_bruta)), length(as.matrix(m_bruta)), sum(m_bruta),
                sum(m_bruta > 0), round(zeros, 1)))
)
tamanho
m_trab <- m_bruta
m_dists <- dist(m_trab, method = "euclidian", diag = TRUE, upper = FALSE)
#m_dists
str(m_dists)
mode(m_dists)
class(m_dists)
length(as.matrix(m_dists))
as.matrix(m_dists)[1:10, 1:10]
range(m_trab)
range(m_dists)
min(m_dists)
max(m_dists)
mean(m_dists) #CENTROIDE!! ou Grand mean
sd(m_dists) #standard deviation
centroide <- mean(m_dists)
centroide
length(m_dists)
m <- nrow(as.matrix(m_dists))
m
m*(m-1)/2
summary(m_dists)
Sumario1 <- cbind(min(m_dists),

```

```

        max(m_dists),
        sd(m_dists),
        mean(m_dists),
        length(m_dists))
colnames(Sumario1) <- c("Minimo", "Maximo", "Desv.Padr", "Media", "m(m-1)/2")
rownames(Sumario1) <- ("Valores")
Sumario1
dist <- (as.matrix(m_dists))
#View(dist)
str(dist)
mode(dist)
class(dist)
#head(dist)
dist[1:5, 1:5] #mostra as 5 primeiras linhas e colunas da matriz
range(m_dists) # intervalo dos valores da matriz
mean(m_dists) # média dos valores da matriz
sd(m_dists) # desvio padrão dos valores da matriz
library(moments)
sk <- skewness(as.matrix(m_dists)) #calcula a assimetria da matriz
ku <- kurtosis(as.matrix(m_dists)) #calcula a curtose da matriz
sku <- cbind(sk,ku) # junta os dois valores em um vetor
colnames(sku) <- c("assimetria", "curtose") # nomeia as colunas do vetor
sku[1:10, ] # mostra as primeiras 10 linhas desse vetor
summary(sku) # mostra um resumo estatístico do vetor
range(m_dists)
par(mfrow=c(2,1)) # esse comando faz os gráficos aparecerem um acima do outro
hist(m_dists,
      breaks = 20, # determina o número de colunas do histograma
      xlim = range(floor(min(m_dists)), ceiling(max(m_dists))), # limites dos eixos
      xlab = "Distr. de Frequências",
      freq = FALSE, # mostra as frequências relativas em vez das absolutas
      main = "Histograma e boxplot da matriz de distâncias")
curve(dnorm(x, mean=mean(m_dists), sd=sd(m_dists)), add=TRUE) # sobrepõe a curva de normalidade
boxplot.default(m_dists, horizontal = TRUE, frame = FALSE,
                xlab="Distr. de Frequências",
                ylim=c(floor(min(m_dists)), ceiling(max(m_dists)))) # limites dos eixos
## dev.off()
library(vegan)
m_relcol <- decostand(m_trab,
                       method="total",
                       MARGIN = 2)
#View(m_relcol)
m_relcol
#colSums(m_relcol)

```

```

#range(m_relcol)
m_relcol_asrq <- asin(sqrt(m_relcol))
#View(m_relcol_asrq)
m_relcol_asrqpi <- 2/pi*(asin(sqrt(m_relcol)))
#View(m_relcol_asrqpi)
#colSums(m_relcol_asrqpi)
#range(m_relcol_asrqpi)
m_trns <- m_relcol_asrqpi
#m_dists <- dist(m_trns, method = "euclidian", diag = TRUE, upper = FALSE)
## m_relcol <- decostand(m_bruta,
##                         method="total",
##                         MARGIN = 2)
## View(m_relcol)
## m_relcol
## m_lg10 <- log10(m_bruta)
## #View(m_lg10)
## m_lg10 <- replace(m_lg10, is.infinite(m_lg10), 0)
## head(m_lg10)
## m_lgx1 <- log(m_bruta+1)
## #View(m_lgx1)
## m_bruta_P1 <- m_bruta^1 #p=1
## #View(m_bruta_P1)
## m_bruta_P05 <- m_bruta^0.5 #p=0,5
## #View(m_bruta_P05)
## m_bruta_P01 <- m_bruta^0.1 #p=0,1
## #View(m_bruta_P01)
## m_pa <- (m_bruta>0)*1L
## #View(m_pa)
## m_rq <- sqrt(m_bruta)
## #View(m_rq)
## m_asrq <- asin(sqrt(m_relcol))
## #View(m_asrq)
## m_asrqpi <- 2/pi*(asin(sqrt(m_relcol)))
## View(m_asrqpi)

```

Referências

Chapter 15

R Módulo 6 - Análise de outliers baseada no desvio padrão do centróide

Apresentação

A análise de outliers baseada no desvio padrão do centróide em análise multivariada é uma técnica utilizada para identificar valores discrepantes em uma matriz de dados com várias variáveis. Essa técnica consiste em calcular o centróide dos dados (ou seja, a média de cada variável), e então calcular o desvio padrão de cada observação em relação ao centróide. Os valores que estão a uma distância maior que um certo número de desvios padrão do centróide são considerados outliers. Essa técnica é útil para identificar observações que podem estar afetando a análise de dados multivariados, como análise de componentes principais, e que podem precisar ser tratadas de forma especial. No entanto, é importante lembrar que a identificação de outliers é uma questão subjetiva e depende do contexto da análise e do objetivo do estudo.

15.1 Sobre os dados do PPBio

Usaremos para esse tutorial dados coletados no Programa de Pesquisa em Biodiversidade - PPBio (Veja Programa de Pesquisa em Biodiversidade – PPBio). Nesta base de dados estão armazenadas informações sobre diversos grupos taxonômicos distribuídos em diversas unidades amostrais (UA's ou sítios), como peixes, macroinvertebrados bentônicos, quironomídeos e zooplâncton, além de dados do habitat, como variáveis físicas e químicas, morfologia do habitat, composição do substrato, estrutura de habitat marginal, entre outros (Figura 15.1).

Essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	sitio	ap-davis	as-bimac	as-fasci	ch-bimac	ci-ocela	ci-orien	co-macro	co-heter	cr-menez	cu-lepid	cy	cy
2	S-A-ZA1	0	1	0	0	0	0	0	0	0	0	0	0
3	S-R-CC1	0	99	0	0	0	0	0	0	0	0	0	0
4	S-R-CT1	0	194	55	0	0	5	0	1	14	0	0	0
5	S-R-CP1	0	19	0	0	0	0	0	0	0	0	0	0
6	S-A-TA1	0	23	1	13	0	0	0	0	0	0	0	0
7	S-R-CT2	0	142	3	3	0	69	0	0	0	4	0	0
8	S-R-CP2	0	5	1	0	40	9	0	0	0	0	0	0
9	S-A-TA2	0	46	0	178	0	0	0	0	0	0	0	0
10	S-R-CT3	0	206	64	0	0	25	0	0	0	8	0	0
11	S-R-CP3	0	16	0	0	13	24	0	0	0	0	0	0
12	S-A-TA3	0	234	7	238	0	0	2	0	0	0	0	0
13	S-R-CT4	0	0	1	0	0	5	0	0	0	1	0	0
14	S-R-CP4	0	0	0	0	11	6	0	0	0	0	0	0
15	S-A-TA4	0	394	0	273	0	0	0	0	0	1	0	0
16	B-A-MU1	0	12	0	0	0	0	0	0	0	0	0	0
17	B-R-ET1	0	3	0	0	0	0	0	0	0	0	0	0
18	B-A-GU1	0	2	2	0	0	0	0	0	0	0	0	0
19	B-R-PC2	5	44	0	0	2	0	0	0	0	0	0	0
20	B-A-MU2	0	99	0	0	0	0	0	0	0	0	0	0
21	B-A-GU2	0	0	0	0	0	0	0	0	0	0	0	0
22	B-R-PC3	22	75	7	0	4	0	0	0	0	0	0	21
23	B-A-MU3	0	511	0	0	0	0	0	0	0	0	0	0
24	B-A-GU3	0	6	0	0	0	0	0	0	0	0	0	0
25	B-R-PC4	0	7	17	0	0	0	0	0	0	0	0	0
26	B-A-MU4	0	235	0	0	0	0	0	0	0	0	0	0
27	B-A-GU4	0	13	0	0	0	0	0	0	0	0	0	0

Figure 15.1: Parte da planilha de dados brutos do PPBio.

para os valores de Captura Por Unidade de Esforço (CPUE), nem foram relativizados ou transformados (Tabela 15.1). As matrizes disponíveis para análises, com suas descrições e tipos de dados recomendados são mostradas na Tabela 15.1.

```
## Warning in table_info$align_vector[column] <-
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para
## para substituir não é um múltiplo do comprimento do substituto
```

A planilha ppbio contém o delineamento amostral de um dos estudos do Projeto PPBio (Figura 15.2). Nas linhas são apresentadas as abreviações dos nomes das unidades amostrais (UA's) e nas colunas são apresentados os nomes abreviados das espécies - temos portanto uma matriz comunitária (15.1). No corpo da planilha temos os valores para o tipo de dados amostrado. Quantitativo, semi-quantitativo ou qualitativo.

Qual desses tipos de dados você acha que é apresentado na planilha?

Várias das espécies nessa matriz tem grande importância ecológica, como é o caso de *Astyanax bimaculatus*¹ (Figura 15.3), que é muito comum em rios intermitentes e serve de alimento para predadores maiores como a espécie *Hoplias malabaricus*² (Figura 15.4).

15.2 Organização básica

```
dev.off() #apaga os graficos, se houver algum
rm(list=ls(all=TRUE)) #limpa a memória
cat("\014") #limpa o console
```

Instalando os pacotes necessários para esse módulo

```
install.packages("openxlsx") #importa arquivos do excel
install.packages("moments") #calcula assimetria e curtose dos dados
install.packages("matrixStats") #fornecce funções rápidas para a estatística de matrizes
install.packages("gt") #ferramenta para criação de tabelas bonitas e personalizáveis
```

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a**

¹A etimologia do gênero *Astyanax* vem da mitologia Grega. Heitor personagem da “Ilíada”, tinha um filho chamado Astianax.

²Do Grego, *hoplon*, arma ou armadura, em referência aos dentes caniniformes muito desenvolvidos, e forte estrutura óssea na cabeça.

Table 15.1: Exemplos de matrizes disponíveis para análises (PPBio Semiárido), com suas descrições e tipos de dados recomendados.

Arquivo (.xlsx)	Tipo de matriz	Descrição	Tipo de dados
ppbio06c	Matriz comunitária	O arquivo ppbio06 traz os dados brutos que serão usados nas análises. A matriz de dados brutos contendo 26 localidades em estações do ano diferentes (objetos) x 35 espécies (atributos), antes de qualquer modificação.	Contagens de indivíduos, alta amplitude de variação sugerido uso de matriz relativizada.
ppbio06h	Matriz ambiental	O arquivo ppbio06h traz os dados brutos que serão usados nas análises. A matriz de dados brutos contendo 26 localidades em estações diferentes (objetos) x 35 variáveis ambientais (atributos) medidas em diferentes escalas espaciais, antes de qualquer modificação.	Unidades de medição das variáveis (cm, m, °C, mg/L, etc.), uma alta amplitude de variação sugerido uso de matriz transformada e/ou reescalação.
ppbio06	Matriz comunitária	O arquivo ppbio06 traz os dados brutos que serão usados nessa análise. A matriz de dados brutos contendo 26 locais/ocasiões (objetos) x 35 espécies (atributos), antes de qualquer modificação.	Contagens de indivíduos, alta amplitude de variação sugerido uso de matriz relativizada.
ppbio06cpue	Matriz comunitária	O arquivo ppbio06cpue traz os valores depois de terem sido ajustados pela Captura Por Unidade de Esforço (CPUE), onde o número de indivíduos de cada espécie em uma determinada UA é dividido pelo esforço de captura daquela UA. Isso significa que os dados foram relativizados pela CPUE. A matriz de dados brutos contendo 26 localidades em estações do ano diferentes (objetos) x 35 espécies (atributos), antes de qualquer modificação.	Densidades de indivíduos de indivíduos por Unidade de Esforço de Captura (densidade relativa), alta amplitude de variação sugerido uso de matriz relativizada.

Figure 15.2: Associação entre a planilha de dados brutos do PPBio e o delineamento amostral do estudo.



Figure 15.3: *Astyanax bimaculatus*, a espécie mais comum da matriz de dados ppbio. Peru, by Eakins, R. Fonte: <<https://www.fishbase.se/summary/Astyanax-bimaculatus.html>>

contra-barra “\”.

Definindo o diretório de trabalho e installando os pacotes necessários:

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

15.2.1 Importando a planilha

Note que o símbolo # em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações.

- Ajuste a primeira linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.
- Ajuste o parâmetro sheet = "Sheet1" para refletir a aba correta do arquivo .xlsx a ser importado.

Alternativamente você pode ir na barra de tarefas e escolhes as opções:
SESSION -> SET WORKING DIRECTORY -> CHOOSE DIRECTORY

```
library(openxlsx)
m_bruta <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06c-p",
                      rowNames = T, colNames = T,
                      sheet = "Sheet1")
str(m_bruta)
m_bruta_ma <- as.matrix(m_bruta) #lê m_bruta como uma matriz
str(m_bruta_ma)
```



Figure 15.4: **Hoplias malabaricus**, espécie que cresce para se tornar um importante predador. Brazil, by Roselet, F.F.G. Fonte: <<https://www.fishbase.se/summary/Hoplias-malabaricus.html>>

```
#m_bruta
m_bruta_ma[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
```

```
## 'data.frame':   26 obs. of  35 variables:
##   $ ap-davis : num  0 0 0 0 0 0 0 0 0 ...
##   $ as-bimac : num  1 99 194 19 23 142 5 46 206 16 ...
##   $ as-fasci : num  0 0 55 0 1 3 1 0 64 0 ...
##   $ ch-bimac : num  0 0 0 0 13 3 0 178 0 0 ...
##   $ ci-ocela : num  0 0 0 0 0 0 40 0 0 13 ...
##   $ ci-orien : num  0 0 5 0 0 69 9 0 25 24 ...
##   $ co-macro : num  0 0 0 0 0 0 0 0 0 ...
##   $ co-heter : num  0 0 1 0 0 0 0 0 0 ...
##   $ cr-menez : num  0 0 14 0 0 4 0 0 8 0 ...
##   $ cu-lepid : num  0 0 0 0 0 0 0 0 0 ...
##   $ cy-gilbe : num  0 0 0 0 0 0 0 0 0 ...
##   $ ge-brasi : num  0 0 3 0 0 0 0 0 1 0 ...
##   $ he-margi : num  0 0 0 0 0 1 0 0 0 0 ...
##   $ ho-malab : num  0 0 1 5 0 17 10 2 31 4 ...
##   $ hy-pusar : num  0 0 9 2 0 43 2 0 11 0 ...
##   $ le-melan : num  0 0 0 0 0 0 0 0 0 ...
##   $ le-piau : num  0 0 3 0 0 1 3 0 2 1 ...
##   $ le-taeni : num  0 0 0 0 0 0 0 0 0 ...
##   $ mo-costa : num  0 0 0 0 0 0 0 0 0 ...
##   $ mo-lepid : num  0 1 39 0 0 1 0 0 0 0 ...
##   $ or-nilot : num  0 2 36 0 0 77 0 0 138 0 ...
##   $ pa-manag : num  0 0 0 0 0 0 0 0 0 ...
##   $ pimel-sp : num  0 0 6 0 0 0 0 0 0 ...
##   $ po-retic : num  0 0 0 0 0 20 0 0 5 0 ...
##   $ po-vivip : num  0 0 47 15 0 221 32 0 326 10 ...
##   $ pr-brevi : num  9 0 5 0 1 15 5 2 164 0 ...
##   $ ps-rhomb : num  0 0 0 0 0 0 0 0 1 0 ...
##   $ ps-genise: num  0 0 0 0 0 0 0 0 1 0 ...
##   $ se-heter : num  0 0 40 14 4 60 0 0 38 0 ...
##   $ se-piaba : num  0 0 68 0 0 0 0 0 0 0 ...
##   $ se-spilo : num  0 0 0 0 0 0 0 0 1 0 ...
##   $ st-noton : num  0 0 1 0 0 25 0 0 115 0 ...
##   $ sy-marmo : num  0 0 0 0 0 1 0 0 0 ...
##   $ te-chalc : num  0 0 0 0 0 0 0 0 0 ...
##   $ tr-signa : num  0 0 18 0 0 15 0 0 7 0 ...
##   num [1:26, 1:35] 0 0 0 0 0 0 0 0 0 0 ...
##   - attr(*, "dimnames")=List of 2
##     ..$ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
##     ..$ : chr [1:35] "ap-davis" "as-bimac" "as-fasci" "ch-bimac" ...
##                   ap-davis as-bimac as-fasci ch-bimac ci-ocela
```

```
## S-A-ZA1      0      1      0      0      0
## S-R-CC1      0     99      0      0      0
## S-R-CT1      0    194     55      0      0
## S-R-CP1      0     19      0      0      0
## S-A-TA1      0     23      1     13      0
```

15.3 Reset point

ATENÇÃO Aqui substitui-se uma nova matriz de dados, relativizada e/ou transformada, pela matriz de trabalho inicial.

```
#m_bruta <- (m_bruta) # <1>
```

1. Aqui usaremos as matrizes relativizadas/transformadas/particionadas, etc

Podemos exibir a planilha depois de ter sido importada para o ambiente R/RStudio usando as funções `View()`, `print()` ou `head()`. Note que essas funções são case-sensitive. A função `ignore.case()` é uma função do pacote `stringr` que modifica um padrão para que ele não considere o caso das letras nas correspondências. Por exemplo, se você quiser encontrar todas as ocorrências da letra “a” em um vetor de caracteres, independente de ser “A” ou “a”, você pode usar essa função.

```
#View(m_bruta)
print(m_bruta)
head(m_bruta)
```

A função `head()` no RStudio é uma forma de ver as primeiras (n=6) linhas de um objeto, como um vetor, uma matriz, um data frame ou uma lista. Ela é útil para ter uma ideia do conteúdo e da estrutura do objeto.

Também podemos explorar as características da planilha usando as funções `str()`, `mode()`, `class()` e `length()`. O número de observações ou tamanho do vetor depende do tipo de dados, se eles são uma `matrix` ou um `data.frame`.

```
str(m_bruta)
mode(m_bruta)
class(m_bruta)
#?str

## 'data.frame':   26 obs. of  35 variables:
## $ ap-davis : num  0 0 0 0 0 0 0 0 0 ...
## $ as-bimac : num  1 99 194 19 23 142 5 46 206 16 ...
## $ as-fasci : num  0 0 55 0 1 3 1 0 64 0 ...
## $ ch-bimac : num  0 0 0 0 13 3 0 178 0 0 ...
## $ ci-ocela : num  0 0 0 0 0 40 0 0 13 ...
```

```

## $ ci-orien : num 0 0 5 0 0 69 9 0 25 24 ...
## $ co-macro : num 0 0 0 0 0 0 0 0 0 0 ...
## $ co-heter : num 0 0 1 0 0 0 0 0 0 0 ...
## $ cr-menez : num 0 0 14 0 0 4 0 0 8 0 ...
## $ cu-lepid : num 0 0 0 0 0 0 0 0 0 0 ...
## $ cy-gilbe : num 0 0 0 0 0 0 0 0 0 0 ...
## $ ge-brasi : num 0 0 3 0 0 0 0 0 1 0 ...
## $ he-margi : num 0 0 0 0 0 1 0 0 0 0 ...
## $ ho-malab : num 0 0 1 5 0 17 10 2 31 4 ...
## $ hy-pusar : num 0 0 9 2 0 43 2 0 11 0 ...
## $ le-melan : num 0 0 0 0 0 0 0 0 0 0 ...
## $ le-piau : num 0 0 3 0 0 1 3 0 2 1 ...
## $ le-taeni : num 0 0 0 0 0 0 0 0 0 0 ...
## $ mo-costa : num 0 0 0 0 0 0 0 0 0 0 ...
## $ mo-lepid : num 0 1 39 0 0 1 0 0 0 0 ...
## $ or-nilot : num 0 2 36 0 0 77 0 0 138 0 ...
## $ pa-manag : num 0 0 0 0 0 0 0 0 0 0 ...
## $ pimel-sp : num 0 0 6 0 0 0 0 0 0 0 ...
## $ po-retic : num 0 0 0 0 0 20 0 0 5 0 ...
## $ po-vivip : num 0 0 47 15 0 221 32 0 326 10 ...
## $ pr-brevi : num 9 0 5 0 1 15 5 2 164 0 ...
## $ ps-rhomb : num 0 0 0 0 0 0 0 0 1 0 ...
## $ ps-genise: num 0 0 0 0 0 0 0 0 1 0 ...
## $ se-heter : num 0 0 40 14 4 60 0 0 38 0 ...
## $ se-piaba : num 0 0 68 0 0 0 0 0 0 0 ...
## $ se-spilo : num 0 0 0 0 0 0 0 0 1 0 ...
## $ st-noton : num 0 0 1 0 0 25 0 0 115 0 ...
## $ sy-marmo : num 0 0 0 0 0 0 1 0 0 0 ...
## $ te-chalc : num 0 0 0 0 0 0 0 0 0 0 ...
## $ tr-signa : num 0 0 18 0 0 15 0 0 7 0 ...
## [1] "list"
## [1] "data.frame"

```

O símbolo `?` é usado para acessar a documentação de uma função ou um pacote no R. Como mostrado acima você pode saber mais sobre a função `str()`, usando o comando `?str`. Isso vai abrir uma página no menu de ajuda com a descrição, os argumentos, os valores de retorno e os exemplos da função `str()`. Você também pode usar o símbolo `?` para obter informações sobre um pacote inteiro. Por exemplo, se você quiser saber mais sobre o pacote `openxlsx`, você pode digitar `?openxlsx`. Isso vai abrir uma página com a visão geral, a instalação, os recursos e as referências do pacote solicitado.

Podemos agora calcular o número e a proporção de zeros na matriz usando as funções `sum()` e `length()` (Você pode pesquisar o que faz a função `length()` usando o comando `?length`).

15.3.1 Tamanho da matriz

```

range(m_bruta) #menor e maior valores
length(m_bruta) #no. de colunas
ncol(m_bruta) #no. de N colunas
nrow(m_bruta) #no. de M linhas
sum(lengths(m_bruta)) #soma os nos. de colunas
length(as.matrix(m_bruta)) #tamanho da matriz m x n
sum(m_bruta == 0) # número de observações igual a zero
sum(m_bruta > 0) # número de observações maiores que zero
zeros <- (sum(m_bruta == 0)/length(as.matrix(m_bruta)))*100 # proporção de zeros na matriz
zeros

## [1] 0 511
## [1] 35
## [1] 35
## [1] 26
## [1] 910
## [1] 910
## [1] 716
## [1] 194
## [1] 78.68132

```

A matriz de dados apresenta um total de 910 valores que variam entre 0, 511 (menor e maior valores). A matriz m x n tem 26 linhas e 35 colunas. Existem 716 observações iguais a zero e 194 observações maiores que zero, representando um percentual de 78.7% dos valores sendo zeros.

Essas informações podem ser resumidas na Tabela 15.2 que será gerada abaixo.

```

tamanho <- data.frame(
  Comando = c("range", "length", "m cols", "n linhas", "Tamanho", "Tamanho",
             "Zeros", "Nao zeros", "% Zeros"),
  Resultado = c(paste(range(m_bruta), collapse = " - "), length(m_bruta), ncol(m_bruta),
                nrow(m_bruta), sum(lengths(m_bruta)), length(as.matrix(m_bruta)), sum(m_bruta ==
sum(m_bruta > 0), round(zeros, 1)))
)
tamanho

##      Comando Resultado
## 1      range    0 - 511
## 2     lenght       35
## 3      m cols      35
## 4     n linhas      26
## 5      Tamanho     910
## 6      Tamanho     910

```

```
## 7      Zeros      716
## 8 Nao zeros     194
## 9 % Zeros      78.7
```

Table 15.2: Resumo das informações sobre o tamanho da matriz

Comando	Resultado
range	0 - 511
length	35
m cols	35
n linhas	26
Tamanho	910
Tamanho	910
Zeros	716
Nao zeros	194
% Zeros	78.7

Ou seja, temos uma matriz de tamanho $m \times n$ igual a 26 objetos por 35 atributos, onde 78.68% dos valores da matriz são iguais a zero!

15.4 Cálculo de matriz de distâncias Euclidiana (`euclid`)

Agora vamos calcular a matriz de **distâncias euclidianas** (BORCARD; GILLET; LEGENDRE, 2018; LEGENDRE; LEGENDRE, 1998) entre as UA's usando a função `dist()` (pesquise o que faz essa função usando o comando `?dist`).

```
euclid <- dist(m_bruta, method = "euclidean", diag = TRUE, upper = FALSE)
##?dist
```

Pronto, calculamos a matriz de distâncias (Euclidiana). Agora podemos visualizar a matriz.

```
#euclid
str(euclid)
mode(euclid)
class(euclid)
length(as.matrix(euclid))
as.matrix(euclid)[1:6, 1:6] #mostra as 5 primeiras linhas e colunas da matriz

##  'dist' num [1:325] 98.4 228.5 29.2 27.1 294.1 ...
##  - attr(*, "Size")= int 26
```

15.4. CÁLCULO DE MATRIZ DE DISTÂNCIAS EUCLIDIANA (EUCLID)321

```

## - attr(*, "Labels")= chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
## - attr(*, "Diag")= logi TRUE
## - attr(*, "Upper")= logi FALSE
## - attr(*, "method")= chr "euclidean"
## - attr(*, "call")= language dist(x = m_bruta, method = "euclidian", diag = TRUE, upper = FALSE)
## [1] "numeric"
## [1] "dist"
## [1] 676
##          S-A-ZA1   S-R-CC1   S-R-CT1   S-R-CP1   S-A-TA1   S-R-CT2
## S-A-ZA1    0.00000 98.43780 228.5235 29.24038 27.09243 294.0629
## S-R-CC1    98.43780  0.00000 154.2433 82.79493 77.25283 261.3905
## S-R-CT1   228.52352 154.24331  0.0000 208.52338 209.69263 224.9133
## S-R-CP1   29.24038  82.79493 208.5234  0.00000 23.25941 271.4922
## S-A-TA1   27.09243  77.25283 209.6926 23.25941  0.00000 283.7869
## S-R-CT2  294.06292 261.39051 224.9133 271.49217 283.78689  0.0000

```

Note que a matriz `euclid` está como uma classe `dist`. Uma matriz do tipo `dist` no R é um objeto que armazena as distâncias entre as linhas de uma matriz ou um data frame. Ela é criada pela função `dist()`, que calcula as distâncias usando diferentes medidas, como “euclidean”, “manhattan”, “canberra”, “binary” ou “minkowski” (HORTON; KLEINMAN, 2015) (Figura 15.3).

Visualizando a matriz de distâncias, observamos que ela:

- é uma matriz quadrada que contém as distâncias entre cada par de elementos do conjunto de dados,
- terá a dimensão $m \times m$ ($m=\text{linhas}$, $n=\text{colunas}$) e,
- e cada elemento da matriz será a distância entre cada par de observações ou objetos.

A **matriz de distâncias é simétrica**, pois a distância entre i e j é igual à distância entre j e i . A matriz de distâncias também tem **diagonal zero**, pois a distância entre uma observação e ela mesma é zero.

```

## Warning in table_info$align_vector[column] <-
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para
## para substituir não é um múltiplo do comprimento do substituto

```

A matriz `euclid` está como uma classe `dist`, mas podemos transformá-la em uma classe `matrix` usando a função `as.matrix()`.

```

euclid_ma <- (as.matrix(euclid))
#View(euclid_ma)
str(euclid_ma)
mode(euclid_ma)
class(euclid_ma)
euclid_ma[1:5, 1:5] #mostra as 5 primeiras linhas e colunas da matriz

```

Table 15.3: Medidas de distância disponíveis na função ‘dist()’ para calcular a distância entre dois pontos ou objetos em um espaço multidimensional.

Distância	Descrição
Euclidiana	Mede a distância entre dois pontos no espaço, seguindo o teorema de Pitágoras. É uma medida de distância direta entre dois pontos.
Manhattan	Também conhecida como distância da cidade, mede a distância como se estivesse dirigindo em uma cidade. A distância é a soma das diferenças absolutas de cada coordenada.
Canberra	Leva em consideração a proporção de diferenças em relação ao tamanho total das variáveis.
Binária	Mede a semelhança entre dois objetos. É 0 quando dois valores são iguais e 1 caso contrário.
Minkowski	É uma generalização da distância euclidiana e da distância de Manhattan. A distância é controlada por um parâmetro P, que determina a ordem da distância.

```

##  num [1:26, 1:26] 0 98.4 228.5 29.2 27.1 ...
##  - attr(*, "dimnames")=List of 2
##    ..$ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
##    ..$ : chr [1:26] "S-A-ZA1" "S-R-CC1" "S-R-CT1" "S-R-CP1" ...
## [1] "numeric"
## [1] "matrix" "array"
##           S-A-ZA1   S-R-CC1   S-R-CT1   S-R-CP1   S-A-TA1
## S-A-ZA1  0.00000  98.43780 228.5235  29.24038 27.09243
## S-R-CC1  98.43780  0.00000 154.2433  82.79493 77.25283
## S-R-CT1 228.52352 154.24331  0.00000 208.52338 209.69263
## S-R-CP1  29.24038  82.79493 208.5234  0.00000 23.25941
## S-A-TA1 27.09243  77.25283 209.6926  23.25941  0.00000

```

15.5. CALCULANDO O CENTRÓIDE DA MATRIZ DE DISTÂNCIAS EUCLID323

15.5 Calculando o centróide da matriz de distâncias euclid

Primeiro algumas informações básicas da matriz

```
range(m_bruta)
range(euclid)
min(euclid)
max(euclid)
mean(euclid) #CENTROIDE!! OU Grand mean
sd(euclid) #Standard deviation
centroide <- mean(euclid)
centroide
```

```
## [1] 0 511
## [1] 16.27882 707.32666
## [1] 16.27882
## [1] 707.3267
## [1] 285.9043
## [1] 172.9708
## [1] 285.9043
```

A função `mean()` calcula a média de todos os valores da matriz de distâncias, ou seja, a média multivariada, que é o centróide. Nesse caso o **centroide assume o valor de 285.9**.

Usamos agora a fórmula $m*(m-1)/2$, onde m é o no. de objetos sendo comparados, para calcular quantas distâncias temos na nossa matriz.

```
length(euclid)
m <- nrow(m_bruta)
m
m*(m-1)/2
summary(euclid)

## [1] 325
## [1] 26
## [1] 325
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    16.28  142.70 242.82 285.90 423.64 707.33
```

Temos então que m é **26 objetos** (ou linhas), e portanto, a matriz de distâncias tem **325 valores**.

Fazemos agora um breve sumário do que foi calculado até agora com base na matriz de distâncias `euclid`.

```

Sumario1 <- cbind(min(euclid),
                    max(euclid),
                    sd(euclid),
                    mean(euclid),
                    length(euclid))
colnames(Sumario1) <- c("Minimo", "Maximo", "Desv.Padr", "Media", "m(m-1)/2")
rownames(Sumario1) <- ("Valores")
Sumario1

##           Minimo   Maximo Desv.Padr     Media m(m-1)/2
## Valores 16.27882 707.3267 172.9708 285.9043      325

```

15.6 Distribuição de frequências da matriz de distâncias euclid

O código abaixo vai plotar um histograma e um boxplot da distribuição de dados armazenada em `euclid`. Também é adicionada uma curva normal teórica ao histograma utilizando a média e o desvio padrão dos dados (Figura 15.5). Atente que o comando `par(mfrow=c(2,1))` define o layout dos gráficos, especificando que serão plotados 2 gráficos em uma coluna. As funções `floor(min(euclid))`, `ceiling(max(euclid))` definem os limites dos graficos pelo valores mínimo e máximo do objeto (`euclid`).

```

range(euclid)

## [1] 16.27882 707.32666

par(mfrow=c(2,1))
hist(euclid,
      breaks = 15, #determina o no. de colunas do histograma
      xlim = range(floor(min(euclid)), ceiling(max(euclid))),
      xlab = "Distr. de Frequências",
      freq = FALSE)
curve(dnorm(x, mean=mean(euclid), sd=sd(euclid)), add=TRUE)
boxplot.default(euclid, horizontal = TRUE, frame = FALSE,
                xlab="Distr. de Frequências",
                ylim=c(floor(min(euclid)), ceiling(max(euclid)))) #Limites do eixo Y

```

O comando abaixo apaga os gráficos, use-o apenas se necessário.

```
dev.off()
```

15.6. DISTRIBUIÇÃO DE FREQUÊNCIAS DA MATRIZ DE DISTÂNCIAS EUCLID325

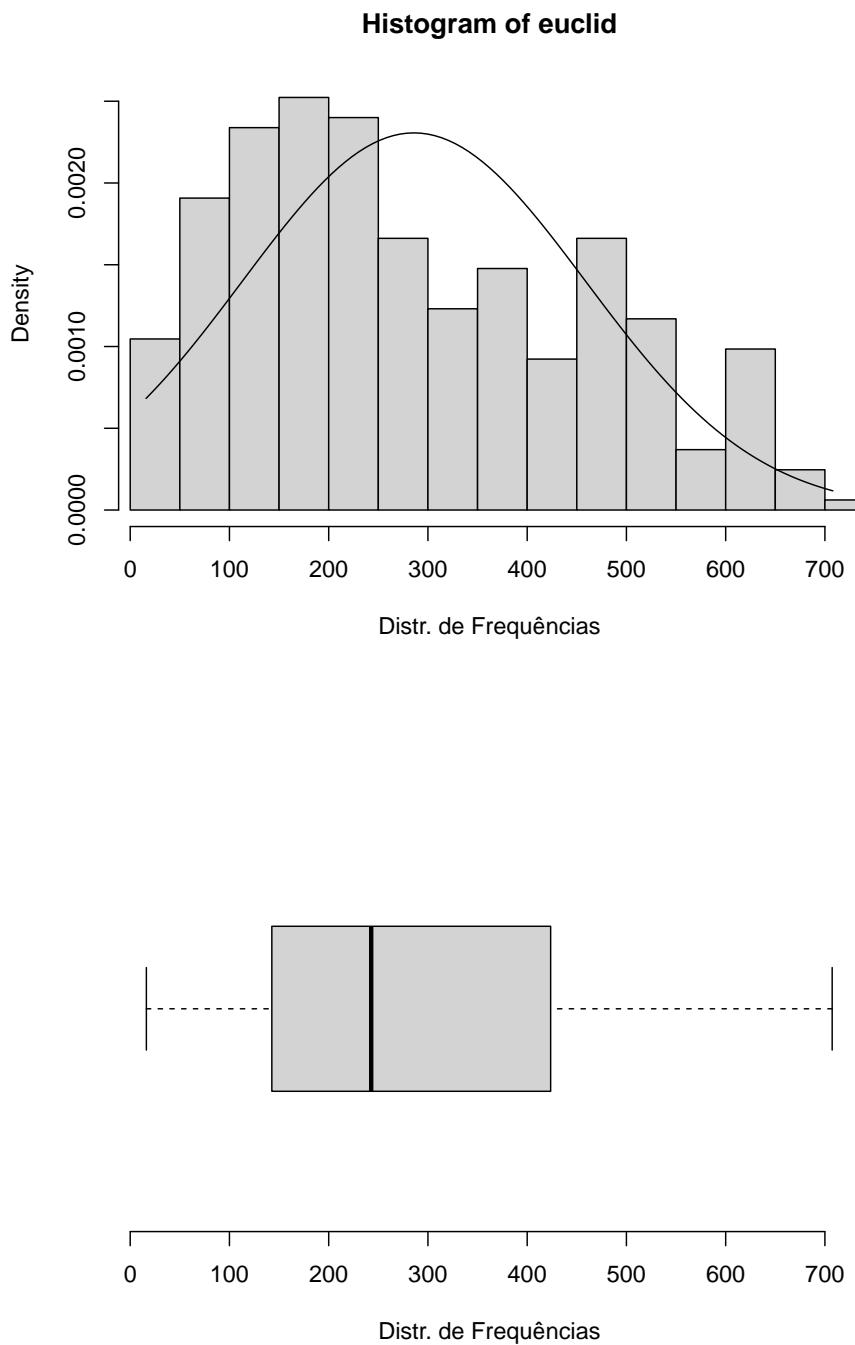


Figure 15.5: Distribuições de frequências da matriz de distâncias `euclid`

15.7 Procurando outliers

15.7.1 Listas de distâncias

Vamos agora criar uma lista com as distâncias em desvio padrão entre cada objeto e o **centróide**, baseada na matriz de distâncias `euclid_ma` (ou outra matriz de distâncias criada a partir de uma matriz relativizada/transformada ou bruta, dependendo do arquivo escolhido). As distâncias médias do centroíde, expressas em desvios padrão, também podem ser chamadas de **z-scores**.

Para esse cálculo é necessário que a matriz de distâncias (do tipo `dist`) seja do tipo `matrix`. Por isso usamos o comando `as.matrix` e removemos a diagonal da nova matriz `euclid_ma`. Códigos a seguir.

15.7.1.1 Removendo os zeros da matriz de distâncias

```
library(matrixStats)

## Warning: package 'matrixStats' was built under R version 4.3.2
##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count

library(moments)
euclid_ma <- as.matrix(euclid)
euclid_ma
range(euclid_ma) #valor errado
mean(euclid_ma) #valor errado
sd(euclid_ma) #valor errado
is.na(euclid_ma) <- euclid_ma==0 #atribui n.a. aos valores = 0
mean(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
mean(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
sd(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
#colMeans(euclid_ma, na.rm=T) #omite valores n.a. do calculo
#rowMeans(euclid_ma, na.rm=T) #omite valores n.a. do calculo

##          S-A-ZA1   S-R-CC1   S-R-CT1   S-R-CP1   S-A-TA1   S-R-CT2   S-R-CP2
## S-A-ZA1    0.00000  98.43780 228.5235  29.24038  27.09243 294.0629  53.40412
## S-R-CC1   98.43780   0.00000 154.2433  82.79493  77.25283 261.3905 108.10180
## S-R-CT1  228.52352 154.24331   0.0000 208.52338 209.69263 224.9133 224.09150
## S-R-CP1   29.24038  82.79493 208.5234   0.00000 23.25941 271.4922 49.22398
## S-A-TA1   27.09243  77.25283 209.6926  23.25941   0.00000 283.7869  57.82733
## S-R-CT2  294.06292 261.39051 224.9133 271.49217 283.78689   0.0000 268.95167
```

```

## S-R-CP2 53.40412 108.10180 224.0915 49.22398 57.82733 268.9517 0.00000
## S-A-TA2 183.74439 185.75791 261.8301 181.24845 166.66133 326.3327 190.15257
## S-R-CT3 460.42155 428.64437 366.6429 443.90877 453.59674 238.3799 438.54532
## S-R-CP3 34.17601 88.06816 215.2557 31.32092 33.13608 275.0436 40.37326
## S-A-TA3 333.86674 274.45765 269.9444 321.82604 309.19735 360.8213 334.61769
## S-R-CT4 187.51533 213.67265 273.0769 189.67604 191.60898 298.0084 190.90835
## S-R-CP4 81.58431 127.98437 226.6208 69.86415 85.24084 238.5351 57.30620
## S-A-TA4 478.64601 402.05970 359.7819 464.33932 453.15560 450.2344 478.08995
## B-A-MU1 190.62529 209.01196 287.5257 191.40272 190.90312 344.8086 197.56012
## B-R-ET1 36.13862 102.02941 228.2740 43.89761 42.52058 291.0859 63.79655
## B-A-GU1 16.27882 97.90301 227.0507 30.36445 28.40775 293.4110 55.07268
## B-R-PC2 91.52595 98.07650 204.8560 86.04650 84.49260 282.9558 103.56158
## B-A-MU2 119.74974 68.23489 165.6351 106.08958 103.04368 262.9848 125.71396
## B-A-GU2 110.95495 147.95607 249.2348 114.19282 113.77610 305.9902 122.87799
## B-R-PC3 151.57506 133.59266 183.3085 135.95588 140.16776 259.3260 157.58490
## B-A-MU3 534.53718 441.89592 367.2261 516.37293 513.70809 451.3635 530.43944
## B-A-GU3 353.04249 363.59180 396.0858 353.76829 353.56612 414.4539 356.90195
## B-R-PC4 89.73294 127.93748 229.5801 87.46999 89.97778 295.9341 103.55675
## B-A-MU4 641.28309 612.29405 595.5023 631.36598 633.67026 590.7597 633.81385
## B-A-GU4 230.36059 244.38903 302.0546 230.92640 230.49512 341.1700 236.11438
## S-A-TA2 S-R-CT3 S-R-CP3 S-A-TA3 S-R-CT4 S-R-CP4 S-A-TA4
## S-A-ZA1 183.7444 460.4215 34.17601 333.8667 187.5153 81.58431 478.6460
## S-R-CC1 185.7579 428.6444 88.06816 274.4576 213.6726 127.98437 402.0597
## S-R-CT1 261.8301 366.6429 215.25566 269.9444 273.0769 226.62083 359.7819
## S-R-CP1 181.2484 443.9088 31.32092 321.8260 189.6760 69.86415 464.3393
## S-A-TA1 166.6613 453.5967 33.13608 309.1973 191.6090 85.24084 453.1556
## S-R-CT2 326.3327 238.3799 275.04363 360.8213 298.0084 238.53511 450.2344
## S-R-CP2 190.1526 438.5453 40.37326 334.6177 190.9084 57.30620 478.0899
## S-A-TA2 0.0000 478.9008 182.86060 198.3053 264.0038 200.93282 360.8130
## S-R-CT3 478.9008 0.0000 449.24826 478.0994 419.5474 411.12529 530.0566
## S-R-CP3 182.8606 449.2483 0.00000 324.5351 190.8612 74.14850 467.2216
## S-A-TA3 198.3053 478.0994 324.53505 0.0000 384.4750 344.01599 164.3624
## S-R-CT4 264.0038 419.5474 190.86121 384.4750 0.0000 195.29977 515.1932
## S-R-CP4 200.9328 411.1253 74.14850 344.0160 195.2998 0.00000 486.2016
## S-A-TA4 360.8130 530.0566 467.22157 164.3624 515.1932 486.20160 0.0000
## B-A-MU1 262.6481 494.5402 192.39023 377.5142 265.3413 207.01208 506.2549
## B-R-ET1 186.4457 461.2114 47.45524 334.1841 190.2735 88.34025 478.2384
## B-A-GU1 183.8777 462.0985 35.22783 333.2807 189.3278 82.22530 477.9676
## B-R-PC2 195.6246 447.5008 90.57041 315.9209 190.5282 122.53571 451.2815
## B-A-MU2 197.8686 427.9603 110.62549 282.7313 222.6971 140.52046 407.6175
## B-A-GU2 214.5623 466.3872 115.52922 352.2528 207.3089 137.12403 491.9075
## B-R-PC3 223.6672 423.6402 147.47542 315.7942 220.0114 170.50220 440.0898
## B-A-MU3 522.9532 505.9802 520.09038 399.2405 563.5140 534.39873 336.9926
## B-A-GU3 397.2719 518.7784 354.25273 483.3363 352.7846 362.13671 590.8849
## B-R-PC4 202.8398 461.7597 94.17006 341.0029 182.5541 119.84991 481.9544
## B-A-MU4 651.0146 647.9599 634.01104 643.0435 659.8833 626.19965 674.7548

```

```

## B-A-GU4 292.6124 476.7599 231.75849 398.4708 265.0623 244.09834 521.8716
##          B-A-MU1   B-R-ET1   B-A-GU1   B-R-PC2   B-A-MU2   B-A-GU2   B-R-PC3
## S-A-ZA1 190.6253  36.13862  16.27882  91.52595 119.74974 110.95495 151.5751
## S-R-CC1 209.0120 102.02941  97.90301  98.07650  68.23489 147.95607 133.5927
## S-R-CT1 287.5257 228.27396 227.05066 204.85605 165.63514 249.23483 183.3085
## S-R-CP1 191.4027  43.89761  30.36445  86.04650 106.08958 114.19282 135.9559
## S-A-TA1 190.9031  42.52058  28.40775  84.49260 103.04368 113.77610 140.1678
## S-R-CT2 344.8086 291.08590 293.41097 282.95583 262.98479 305.99020 259.3260
## S-R-CP2 197.5601  63.79655  55.07268 103.56158 125.71396 122.87799 157.5849
## S-A-TA2 262.6481 186.44570 183.87768 195.62464 197.86864 214.56235 223.6672
## S-R-CT3 494.5402 461.21145 462.09847 447.50084 427.96028 466.38718 423.6402
## S-R-CP3 192.3902  47.45524  35.22783  90.57041 110.62549 115.52922 147.4754
## S-A-TA3 377.5142 334.18408 333.28066 315.92088 282.73132 352.25275 315.7942
## S-R-CT4 265.3413 190.27349 189.32776 190.52821 222.69710 207.30895 220.0114
## S-R-CP4 207.0121  88.34025  82.22530 122.53571 140.52046 137.12403 170.5022
## S-A-TA4 506.2549 478.23843 477.96757 451.28151 407.61747 491.90751 440.0898
## B-A-MU1  0.0000 193.24078 183.63823 201.66556 151.28781 198.33557 225.3242
## B-R-ET1 193.2408  0.00000 36.53765  97.11334 119.98333 112.60107 151.2977
## B-A-GU1 183.6382  36.53765  0.00000 91.72786 115.33863 98.15294 149.6529
## B-R-PC2 201.6656  97.11334  91.72786  0.00000 114.92171 141.58390 114.7606
## B-A-MU2 151.2878 119.98333 115.33863 114.92171 0.00000 153.39817 143.1049
## B-A-GU2 198.3356 112.60107 98.15294 141.58390 153.39817 0.00000 173.1011
## B-R-PC3 225.3242 151.29772 149.65293 114.76062 143.10486 173.10113 0.0000
## B-A-MU3 505.4424 530.61097 531.72831 497.81221 422.87823 539.86572 476.5574
## B-A-GU3 381.5508 348.29729 342.49380 360.00556 364.76705 257.95736 337.1735
## B-R-PC4 205.1536  95.01579  89.16838 122.22520 142.70249 139.05035 150.8211
## B-A-MU4 498.8387 627.17462 635.41089 625.57414 550.83754 632.75193 618.9628
## B-A-GU4 289.1885 227.41592 219.20538 242.81886 251.58299 129.34837 239.0000
##          B-A-MU3   B-A-GU3   B-R-PC4   B-A-MU4   B-A-GU4
## S-A-ZA1 534.5372 353.0425  89.73294 641.2831 230.3606
## S-R-CC1 441.8959 363.5918 127.93748 612.2940 244.3890
## S-R-CT1 367.2261 396.0858 229.58005 595.5023 302.0546
## S-R-CP1 516.3729 353.7683  87.46999 631.3660 230.9264
## S-A-TA1 513.7081 353.5661  89.97778 633.6703 230.4951
## S-R-CT2 451.3635 414.4539 295.93411 590.7597 341.1700
## S-R-CP2 530.4394 356.9019 103.55675 633.8139 236.1144
## S-A-TA2 522.9532 397.2719 202.83984 651.0146 292.6124
## S-R-CT3 505.9802 518.7784 461.75968 647.9599 476.7599
## S-R-CP3 520.0904 354.2527  94.17006 634.0110 231.7585
## S-A-TA3 399.2405 483.3363 341.00293 643.0435 398.4708
## S-R-CT4 563.5140 352.7846 182.55410 659.8833 265.0623
## S-R-CP4 534.3987 362.1367 119.84991 626.1996 244.0983
## S-A-TA4 336.9926 590.8849 481.95435 674.7548 521.8716
## B-A-MU1 505.4424 381.5508 205.15360 498.8387 289.1885
## B-R-ET1 530.6110 348.2973  95.01579 627.1746 227.4159
## B-A-GU1 531.7283 342.4938  89.16838 635.4109 219.2054

```

```

## B-R-PC2 497.8122 360.0056 122.22520 625.5741 242.8189
## B-A-MU2 422.8782 364.7670 142.70249 550.8375 251.5830
## B-A-GU2 539.8657 257.9574 139.05035 632.7519 129.3484
## B-R-PC3 476.5574 337.1735 150.82109 618.9628 239.0000
## B-A-MU3 0.0000 624.8288 534.66812 519.9981 566.2808
## B-A-GU3 624.8288 0.0000 357.35557 707.3267 134.3763
## B-R-PC4 534.6681 357.3556 0.00000 641.2628 241.6402
## B-A-MU4 519.9981 707.3267 641.26282 0.0000 669.3982
## B-A-GU4 566.2808 134.3763 241.64023 669.3982 0.0000
## [1] 0.0000 707.3267
## [1] 274.908
## [1] 178.1842
## [1] 285.9043
## [1] 285.9043
## [1] 172.8375

```

15.7.1.2 Criando uma Lista

```

centroide_ma <- mean(euclid_ma, na.rm=T)
av.dist <- (as.matrix(colMeans(euclid_ma, na.rm=T)))
av.desvpad <- (as.matrix(colSds(euclid_ma, na.rm=T)))
dp.centroide_ma <- (av.dist-centroide_ma)/(colSds(av.dist)) #ou z-scores
list <- as.matrix(cbind(av.dist, av.desvpad, dp.centroide_ma))
list

##          [,1]      [,2]      [,3]
## S-A-ZA1 202.2608 175.36143 -0.74136373
## S-R-CC1 205.9912 142.40233 -0.70830031
## S-R-CT1 266.3790  93.46194 -0.17306037
## S-R-CP1 195.7828 170.97514 -0.79878039
## S-A-TA1 195.8495 170.39163 -0.79818963
## S-R-CT2 317.0479  82.87727  0.27603716
## S-R-CP2 207.1435 167.09006 -0.69808647
## S-A-TA2 268.5172 124.64348 -0.15410841
## S-R-CT3 455.6678  68.84928  1.50467759
## S-R-CP3 199.1922 170.79087 -0.76856181
## S-A-TA3 347.0119  92.95522  0.54161944
## S-R-CT4 280.9253 131.44373 -0.04413038
## S-R-CP4 217.3521 157.62618 -0.60760370
## S-A-TA4 458.7988  93.50892  1.53242940
## B-A-MU1 278.0482 116.17286 -0.06963167
## B-R-ET1 205.3272 169.74682 -0.71418525
## B-A-GU1 200.2219 174.32564 -0.75943496
## B-R-PC2 215.0274 152.54520 -0.62820821
## B-A-MU2 210.8910 128.79692 -0.66487097

```

330 CHAPTER 15. R MÓDULO 6 - ANÁLISE DE OUTLIERS BASEADA NO DESVIO PADRÃO DO C

```
## B-A-GU2 229.0481 151.90782 -0.50393825
## B-R-PC3 235.2979 130.07777 -0.44854354
## B-A-MU3 499.5754 64.70789 1.89384716
## B-A-GU3 394.6795 116.02951 0.96411611
## B-R-PC4 225.0953 158.58714 -0.53897264
## B-A-MU4 624.1237 45.46670 2.99776610
## B-A-GU4 298.2560 130.86907 0.10947773
```

Ordenamos as distâncias da maior para a menor.

```
colnames(list, do.NULL = FALSE)
colnames(list) <- c("Av.Dist", "Av.StDev", "DP.Centroide")
list2 <- list[order(list[,1], decreasing = TRUE),] #[,1] ou o nome da coluna
list2
```

```
## [1] "col1" "col2" "col3"
##          Av.Dist Av.StDev DP.Centroide
## B-A-MU4 624.1237 45.46670 2.99776610
## B-A-MU3 499.5754 64.70789 1.89384716
## S-A-TA4 458.7988 93.50892 1.53242940
## S-R-CT3 455.6678 68.84928 1.50467759
## B-A-GU3 394.6795 116.02951 0.96411611
## S-A-TA3 347.0119 92.95522 0.54161944
## S-R-CT2 317.0479 82.87727 0.27603716
## B-A-GU4 298.2560 130.86907 0.10947773
## S-R-CT4 280.9253 131.44373 -0.04413038
## B-A-MU1 278.0482 116.17286 -0.06963167
## S-A-TA2 268.5172 124.64348 -0.15410841
## S-R-CT1 266.3790 93.46194 -0.17306037
## B-R-PC3 235.2979 130.07777 -0.44854354
## B-A-GU2 229.0481 151.90782 -0.50393825
## B-R-PC4 225.0953 158.58714 -0.53897264
## S-R-CP4 217.3521 157.62618 -0.60760370
## B-R-PC2 215.0274 152.54520 -0.62820821
## B-A-MU2 210.8910 128.79692 -0.66487097
## S-R-CP2 207.1435 167.09006 -0.69808647
## S-R-CC1 205.9912 142.40233 -0.70830031
## B-R-ET1 205.3272 169.74682 -0.71418525
## S-A-ZA1 202.2608 175.36143 -0.74136373
## B-A-GU1 200.2219 174.32564 -0.75943496
## S-R-CP3 199.1922 170.79087 -0.76856181
## S-A-TA1 195.8495 170.39163 -0.79818963
## S-R-CP1 195.7828 170.97514 -0.79878039
```

15.7.2 Distribuição de frequências das distâncias médias para o centróide av.dist

Observe o valor maximo e minimo gerados pela função `range()` e substitua nas linhas de código assinaladas com `#`. Aqui o menor valor foi 195.7828431 e o maior valor foi 624.1236964. Use valores maiores para facilitar a visualização no gráfico (Figura 15.6).

```
par(mfrow=c(3,1))
hist(list2[, "Av.Dist"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias (em dp) para o centroide",
      main = "Distribuição de Frequência da distância média para o centroide",
      xlim = range(floor(min(av.dist)), ceiling(max(av.dist))), #substitua aqui o menor e maior valor
      freq = T)
hist(list2[, "Av.Dist"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias (em dp) para o centroide",
      main = "Curva de normalidade ajustada para a Distribuição de Frequência",
      xlim = range(floor(min(av.dist)), ceiling(max(av.dist))), #substitua aqui o menor e maior valor
      freq = F)
curve(dnorm(x, mean=mean(list2[, "Av.Dist"]), sd=sd(list2[, "Av.Dist"])), add=TRUE)
boxplot.default(list2[, "Av.Dist"], horizontal = TRUE, frame = FALSE,
                xlab="Distr. de Frequências",
                ylim=c(floor(min(av.dist)), ceiling(max(av.dist)))) #substitua aqui o menor e maior valor

par(mfrow=c(1,1))
```

15.7.2.1 Distribuição de frequências dos desvios padões das distâncias médias para o centróide DP.Centroide

Observe o valor máximo e mínimo gerados pela função `range()` e substitua nas linhas de código assinaladas com `#`. Aqui o menor valor foi -0.7987804 e o maior valor foi 2.9977661. Use valores maiores para facilitar a visualização no gráfico (Figura 15.7).

```
range(dp.centroide_ma)
par(mfrow=c(3,1))
hist(list2[, "DP.Centroide"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias dos desvios padões para o centroide",
      main = "Distribuição de Frequência dos desvio padões das distâncias médias para o centroide",
      xlim = range(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma))), #substitua aqui o menor e maior valor
      freq = T)
hist(list2[, "DP.Centroide"],
```

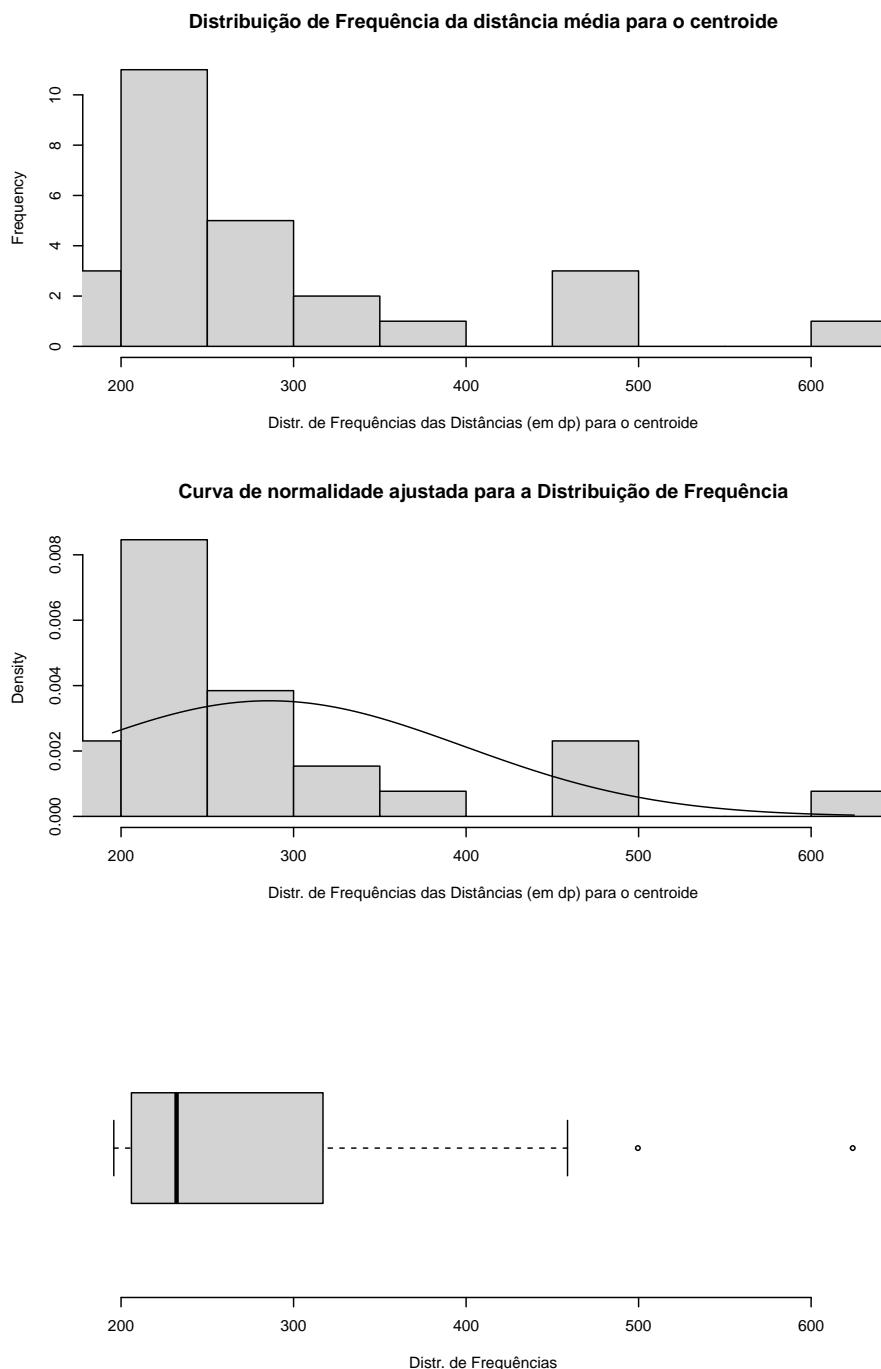


Figure 15.6: Distribuição de frequências das distâncias médias para o centroide.

```

breaks = 15, #determina o no. de colunas do histograma
xlab = "Distr. de Frequências das Distâncias dos desvios padrões das distâncias médias para
main = "Curva de normalidade ajustada para a Distribuição de Frequência",
xlim = range(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma))), #substitua aqui o
freq = F)
curve(dnorm(x, mean=mean(list2[, "DP.Centroide"])), sd=sd(list2[, "DP.Centroide"])), add=TRUE)
boxplot.default(list2[, "DP.Centroide"], horizontal = TRUE, frame = FALSE,
               xlab="Distr. de Frequências",
               ylim=c(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma)))) #substitua aí

par(mfrow=c(1,1))

```

```
## [1] -0.7987804 2.9977661
```

Se necessário apague os gráficos

```
dev.off()
```

15.8 Lista final de outliers

Agora fazemos a lista final de distâncias com os outliers baseados em um ‘cutoff’.

```
cutoff <- 2.0
```

Note que o ‘cutoff’ foi estabelecido no código acima no vetor ‘cutoff <-’. Ou seja, o ‘cutoff’ definindo quem serão considerados outliers, foi estabelecido como sendo valores de **+2.0** e **-2.0** desvios padrões da média multivariada ou centroide. Valores acima ou abaixo do ‘cutoff’ definido recebem o nome “OUT”, na coluna “Outliers”. Criada com o código abaixo.

```

library(gt)
format(cutoff, nsmall = 1)
listf <- as.data.frame(list2)
listf$Outliers <- ifelse(listf$DP.Centroide>-cutoff #CUTOFF MENOR QUE '-cutoff'
                           & listf$DP.Centroide<cutoff, #CUTOFF MAIOR QUE 'cutoff'
                           "", "OUT")
listf
gt(cbind(Sitios=rownames(listf),listf))

## [1] "2.0"
##          Av.Dist  Av.StDev DP.Centroide Outliers
## B-A-MU4 624.1237 45.46670  2.99776610      OUT
## B-A-MU3 499.5754 64.70789  1.89384716

```

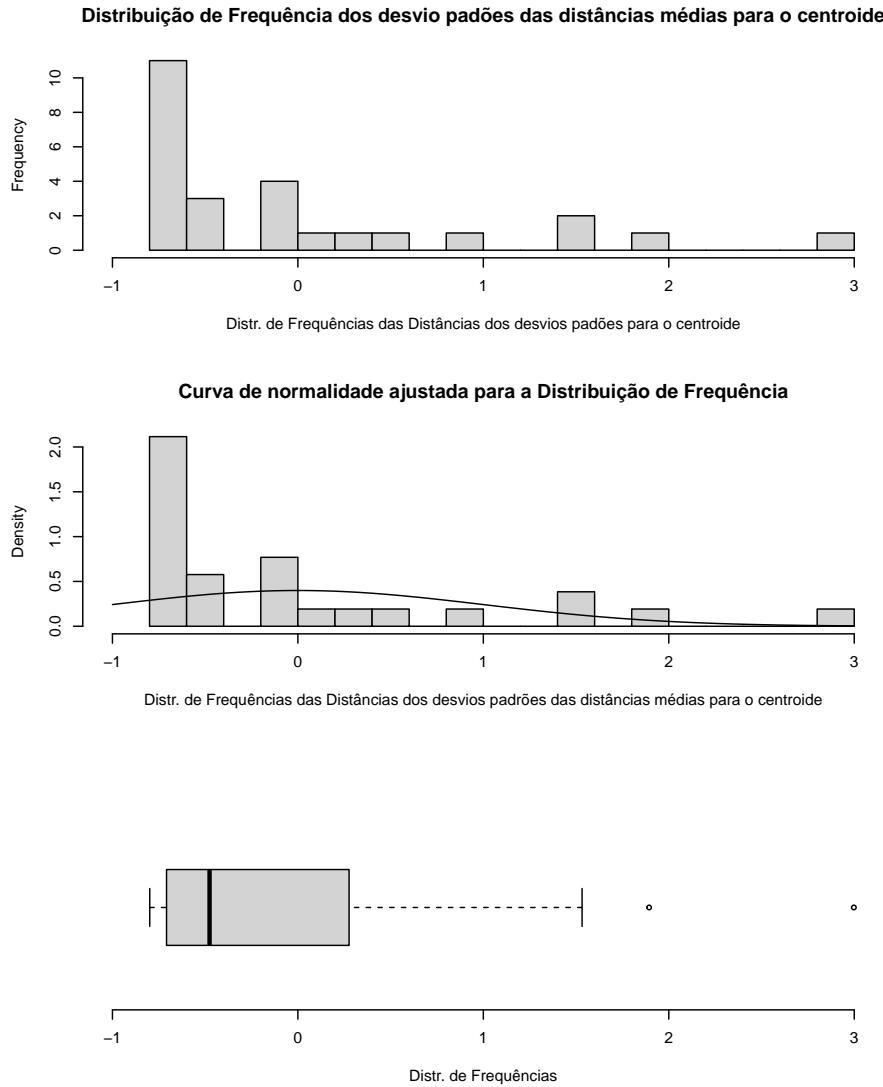


Figure 15.7: Distribuição de frequências dos desvios padrões das distâncias médias para o centroíde.

```

## S-A-TA4 458.7988  93.50892  1.53242940
## S-R-CT3 455.6678  68.84928  1.50467759
## B-A-GU3 394.6795 116.02951  0.96411611
## S-A-TA3 347.0119  92.95522  0.54161944
## S-R-CT2 317.0479  82.87727  0.27603716
## B-A-GU4 298.2560 130.86907  0.10947773
## S-R-CT4 280.9253 131.44373 -0.04413038
## B-A-MU1 278.0482 116.17286 -0.06963167
## S-A-TA2 268.5172 124.64348 -0.15410841
## S-R-CT1 266.3790  93.46194 -0.17306037
## B-R-PC3 235.2979 130.07777 -0.44854354
## B-A-GU2 229.0481 151.90782 -0.50393825
## B-R-PC4 225.0953 158.58714 -0.53897264
## S-R-CP4 217.3521 157.62618 -0.60760370
## B-R-PC2 215.0274 152.54520 -0.62820821
## B-A-MU2 210.8910 128.79692 -0.66487097
## S-R-CP2 207.1435 167.09006 -0.69808647
## S-R-CC1 205.9912 142.40233 -0.70830031
## B-R-ET1 205.3272 169.74682 -0.71418525
## S-A-ZA1 202.2608 175.36143 -0.74136373
## B-A-GU1 200.2219 174.32564 -0.75943496
## S-R-CP3 199.1922 170.79087 -0.76856181
## S-A-TA1 195.8495 170.39163 -0.79818963
## S-R-CP1 195.7828 170.97514 -0.79878039

```

Sitios	Av.Dist	Av.StDev	DP.Centroide	Outliers
B-A-MU4	624.1237	45.46670	2.99776610	OUT
B-A-MU3	499.5754	64.70789	1.89384716	
S-A-TA4	458.7988	93.50892	1.53242940	
S-R-CT3	455.6678	68.84928	1.50467759	
B-A-GU3	394.6795	116.02951	0.96411611	
S-A-TA3	347.0119	92.95522	0.54161944	
S-R-CT2	317.0479	82.87727	0.27603716	
B-A-GU4	298.2560	130.86907	0.10947773	
S-R-CT4	280.9253	131.44373	-0.04413038	
B-A-MU1	278.0482	116.17286	-0.06963167	
S-A-TA2	268.5172	124.64348	-0.15410841	
S-R-CT1	266.3790	93.46194	-0.17306037	
B-R-PC3	235.2979	130.07777	-0.44854354	
B-A-GU2	229.0481	151.90782	-0.50393825	
B-R-PC4	225.0953	158.58714	-0.53897264	
S-R-CP4	217.3521	157.62618	-0.60760370	
B-R-PC2	215.0274	152.54520	-0.62820821	
B-A-MU2	210.8910	128.79692	-0.66487097	
S-R-CP2	207.1435	167.09006	-0.69808647	

S-R-CC1	205.9912	142.40233	-0.70830031
B-R-ET1	205.3272	169.74682	-0.71418525
S-A-ZA1	202.2608	175.36143	-0.74136373
B-A-GU1	200.2219	174.32564	-0.75943496
S-R-CP3	199.1922	170.79087	-0.76856181
S-A-TA1	195.8495	170.39163	-0.79818963
S-R-CP1	195.7828	170.97514	-0.79878039

15.9 Particionando a matriz

Talvez seja necessário particionar a matriz bruta para remover os outliers. Isso pode ser feito com os códigos abaixo. Atente para o vetor `part`. Ele define quais objetos ou UA's serão particionados.

```
part <- c("B-A-MU4", "B-A-MU3", "S-A-TA4", "S-R-CT3")
part

## [1] "B-A-MU4" "B-A-MU3" "S-A-TA4" "S-R-CT3"

m_bruta_part <- m_bruta[!(row.names(m_bruta) %in% c(part)),]
#m_bruta_part
```

Criamos uma nova matriz particionada `m_bruta_part` sem **B-A-MU4**, **B-A-MU3**, **S-A-TA4**, **S-R-CT3**, que apresentavam valores acima do cutoff para outliers.

ATENÇÃO

Quando deletamos linhas ou colunas da matriz de dados comunitária, precisamos ver se isso não gerou linhas ou colunas vazias. Observe por exemplo a espécie *Apareiodon davisi*³ (Figura 15.8). Por exemplo, se uma determinada espécie só ocorre em uma UA, e aquela UA foi removida, essa espécie não consta mais em nenhuma UA e por isso a sua soma vai ser igual a zero. Isso aconteceu com a espécie *Pseudancistrus genisetiger*⁴ (Figura 15.9). Essa espécie precisará ser removida.

Use o a função `View(m_bruta_part)` e procure linhas ou colunas na matriz `m_bruta_part` onde todos os valores são zero. Quais espécies você encontrou que deixaram de ocorrer na comunidade depois que as UA's outliers foram removidas?

Complicado? Os códigos abaixo resolvem isso, e mostram algumas propriedades da nova matriz `m_bruta_part2` depois de particionada pela segunda vez para

³A etimologia do nome *Apareiodon* vem do Grego, *a*, sem, *pareia*, lateral ou bochecha, *odous* dentição, em referência a ausência de tentes laterais no aparato bucal dessa espécie.

⁴A etimologia do nome *Pseudancistrus* vem do Grego, *pseudes*, falso, e *agkistron*, gancho, em referência a falsos ganchos presentes na cabeça em algumas espécies do gênero.



Figure 15.8: **Apareiodon davisi**, importante espécie bentopelágica das bacias dos rios Jaguaribe e Paraíba. Brazil, by Ramos, T.P.A. Fonte: <<https://www.fishbase.se/summary/Astianax-bimaculatus.html>>



Figure 15.9: **Pseudancistrus genisetiger**, uma espécie endêmica das bacias hidrográficas do nordeste. By eplanetcatfish.com. Fonte: <<https://www.planetcatfish.com>> (Provável identificação errada do espécime da foto)](eplanetcatfish.jpg)

remover linhas/colunas “zeradas”.

```

sum <- colSums(m_bruta_part)
sum
zero_sum_cols <- names(which(colSums(m_bruta_part) == 0))
zero_sum_cols #nomes das espécies zeradas
m_bruta_part2 <- m_bruta_part[(colSums(m_bruta_part) != 0)] #em != a exclamação inverte o sentido
zero_sum_cols2 <- names(which(colSums(m_bruta_part2) == 0))
zero_sum_cols2 #nomes das espécies zeradas
sum<-colSums(m_bruta_part2)
sum
#m_bruta_part2
#m_bruta_part2 <- as.matrix(m_bruta_part2)
str(m_bruta_part2)
length(as.matrix(m_bruta_part2))

##   ap-davis   as-bimac   as-fasci   ch-bimac   ci-ocela   ci-orien   co-macro   co-heter
##      27       1040        94       432        70       118         2         1
##   cr-menez   cu-lepid   cy-gilbe   ge-brasi   he-margi   ho-malab   hy-pusar   le-melan
##      19        21       131       364        2        69        60        2
##   le-piau   le-taeni   mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
##      11        1         1       41       697       554         6       64
##   po-vivip  pr-brevi  ps-rhomb  ps-genise  se-heter  se-piaba  se-spilo  st-noton
##      441       111        0         0       258        68         0       90
##   sy-marmo  te-chalc  tr-signa
##      1       134       201
## [1] "ps-rhomb" "ps-genise" "se-spilo"
## character(0)
## ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro co-heter
##      27       1040        94       432        70       118         2         1
##   cr-menez   cu-lepid   cy-gilbe   ge-brasi   he-margi   ho-malab   hy-pusar   le-melan
##      19        21       131       364        2        69        60        2
##   le-piau   le-taeni   mo-costa   mo-lepid   or-nilot   pa-manag   pimel-sp   po-retic
##      11        1         1       41       697       554         6       64
##   po-vivip  pr-brevi  se-heter  se-piaba  st-noton  sy-marmo  te-chalc  tr-signa
##      441       111       258        68        90         1       134       201
## 'data.frame': 22 obs. of 32 variables:
## $ ap-davis: num 0 0 0 0 0 0 0 0 0 ...
## $ as-bimac: num 1 99 194 19 23 142 5 46 16 234 ...
## $ as-fasci: num 0 0 55 0 1 3 1 0 0 7 ...
## $ ch-bimac: num 0 0 0 0 13 3 0 178 0 238 ...
## $ ci-ocela: num 0 0 0 0 0 40 0 13 0 ...
## $ ci-orien: num 0 0 5 0 0 69 9 0 24 0 ...
## $ co-macro: num 0 0 0 0 0 0 0 0 0 2 ...
## $ co-heter: num 0 0 1 0 0 0 0 0 0 0 ...

```

```

## $ cr-menez: num  0 0 14 0 0 4 0 0 0 0 ...
## $ cu-lepid: num  0 0 0 0 0 0 0 0 0 0 ...
## $ cy-gilbe: num  0 0 0 0 0 0 0 0 0 0 ...
## $ ge-brasi: num  0 0 3 0 0 0 0 0 0 0 ...
## $ he-margi: num  0 0 0 0 0 1 0 0 0 0 ...
## $ ho-malab: num  0 0 1 5 0 17 10 2 4 20 ...
## $ hy-pusar: num  0 0 9 2 0 43 2 0 0 0 ...
## $ le-melan: num  0 0 0 0 0 0 0 0 0 0 ...
## $ le-piau : num  0 0 3 0 0 1 3 0 1 0 ...
## $ le-taeni: num  0 0 0 0 0 0 0 0 0 0 ...
## $ mo-costa: num  0 0 0 0 0 0 0 0 0 0 ...
## $ mo-lepid: num  0 1 39 0 0 1 0 0 0 0 ...
## $ or-nilot: num  0 2 36 0 0 77 0 0 0 0 ...
## $ pa-manag: num  0 0 0 0 0 0 0 0 0 0 ...
## $ pimel-sp: num  0 0 6 0 0 0 0 0 0 0 ...
## $ po-retic: num  0 0 0 0 0 20 0 0 0 0 ...
## $ po-vivip: num  0 0 47 15 0 221 32 0 10 0 ...
## $ pr-brevi: num  9 0 5 0 1 15 5 2 0 0 ...
## $ se-heter: num  0 0 40 14 4 60 0 0 0 0 ...
## $ se-piaba: num  0 0 68 0 0 0 0 0 0 0 ...
## $ st-noton: num  0 0 1 0 0 25 0 0 0 0 ...
## $ sy-marmo: num  0 0 0 0 0 0 1 0 0 0 ...
## $ te-chalc: num  0 0 0 0 0 0 0 0 0 0 ...
## $ tr-signa: num  0 0 18 0 0 15 0 0 0 0 ...
## [1] 704

```

15.10 Exportando a matrix final de trabalho

Agora vamos exportar a matrix de trabalho, nesse caso a matrix `m_bruta_part2`, como um arquivo de valores separados por vírgula (`m_bruta_part2.csv.csv`). Na verdade, separado por ";", como definido na função `sep = ";"`. O arquivo será exportado para o diretório de trabalho. Na mesma sequência de códigos, importamos `m_bruta_part2.csv.csv` como nossa matriz de trabalho `m_trab`.

```

df <- data.frame(Sites = rownames(m_bruta),
                  row.names = NULL,
                  check.names = FALSE) #add titulo a primeira coluna

write.table(m_bruta_part2, "m_trabcsv.csv",
            sep = ";", dec = ".", "#"\t",
            row.names = TRUE,
            quote = TRUE,
            append = FALSE)

```

```
m_trab <- read.csv("m_trabcsv.csv",
                     sep = ";", dec = ".",
                     row.names = 1,
                     header = TRUE,
                     na.strings = NA,
                     check.names = FALSE, #impede que o R mude os nomes das colunas
                     col.names = gsub("(^|_)([a-z])", "\\\\1\\\\U\\\\2",
                                      names(m_trab), perl = TRUE))
```

- Inicial Maiuscula: col.names = gsub("(^|_)([a-z])", "\\\\1\\\\U\\\\2", names(m_trab), perl = TRUE))
- Substituição: col.names = gsub("-", ".", names(m_trab)))
- A função **header=** especifica se o arquivo de entrada contém uma linha de cabeçalho com os nomes das colunas, **row.names** especifica qual coluna no arquivo de entrada deve ser usada como os nomes das linhas, e se **NULL**, os nomes das linhas são gerados como números

Agora é com você... Refaça toda a análise com a matriz bruta particionada **m_bruta_part2**

Apêndices

Outras formas de fazer a partição

```
m_bruta_part2 <- m_bruta_part
```

$$\text{colSums}(\text{abs}(m_b ruta_p art), na.rm = F) 0$$

```
m_bruta_part2 <- subset(m_bruta_part, colSums != 0) m_bruta_part2 <-
m_bruta_part
, colSums(m_b ruta_p art! = 0) 0
```

Código

```
range(euclidma) par(mfrow=c(2,1)) hist(euclidma, breaks = 15, #determina o no. de colunas do histograma xlim = range(floor(min(euclidma)), ceiling(max(euclidma))), xlab = "Distr. de Frequências", freq = FALSE) curve(dnorm(x, mean=mean(euclidma), sd=sd(euclidma)), add=TRUE) boxplot.default(euclidma, horizontal = TRUE, frame = FALSE, xlab="Distr. de Frequências", ylim=c(floor(min(euclidma)), ceiling(max(euclidma))))
```

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executá-lo. Lembre de remover os # ou ## caso necessite executar essas linhas.

```
## dev.off() #apaga os graficos, se houver algum
## rm(list=ls(all=TRUE)) #limpa a memória
## cat("\014") #limpa o console
## install.packages("openxlsx") #importa arquivos do excel
## install.packages("moments") #calcula assimetria e curtose dos dados
## install.packages("matrixStats") #fornecce funções rápidas para a estatística de matrizes
## install.packages("gt") #ferramenta para criação de tabelas bonitas e personalizáveis
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
library(openxlsx)
m_bruta <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/peixes06.xlsx",
                      rowNames = T, colNames = T,
                      sheet = "Sheet1")
str(m_bruta)
m_bruta_ma <- as.matrix(m_bruta) #lê m_bruta como uma matriz
str(m_bruta_ma)
#m_bruta
m_bruta_ma[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
#m_bruta <- (m_bruta) # <1>
#View(m_bruta)
print(m_bruta)
head(m_bruta)
str(m_bruta)
mode(m_bruta)
class(m_bruta)
#?str
range(m_bruta) #menor e maior valores
length(m_bruta) #no. de colunas
ncol(m_bruta) #no. de N colunas
nrow(m_bruta) #no. de M linhas
sum(lengths(m_bruta)) #soma os nos. de colunas
length(as.matrix(m_bruta)) #tamanho da matriz m x n
sum(m_bruta == 0) # número de observações igual a zero
sum(m_bruta > 0) # número de observações maiores que zero
zeros <- (sum(m_bruta == 0)/length(as.matrix(m_bruta)))*100 # proporção de zeros na matriz
zeros
tamanho <- data.frame(
  Função = c("range", "length", "m cols", "n linhas", "Tamanho", "Tamanho",
            "Zeros", "Nao zeros", "% Zeros"),
  ...)
```

```
Resultado = c(paste(range(m_bruta), collapse = " - "), length(m_bruta), ncol(m_bruta),
             nrow(m_bruta), sum(lengths(m_bruta)), length(as.matrix(m_bruta)), sum(m_bruta ==
             sum(m_bruta > 0), round(zeros, 1))
)
tamanho
knitr::kable(tamanho, format = "markdown", caption = "Resumo das informações sobre o tamanho da m
euclid <- dist(m_bruta, method = "euclidean", diag = TRUE, upper = FALSE)
#?dist
#euclid
str(euclid)
mode(euclid)
class(euclid)
length(as.matrix(euclid))
as.matrix(euclid)[1:6, 1:6] #mostra as 5 primeiras linhas e colunas da matriz
euclid_ma <- (as.matrix(euclid))
#View(euclid_ma)
str(euclid_ma)
mode(euclid_ma)
class(euclid_ma)
euclid_ma[1:5, 1:5] #mostra as 5 primeiras linhas e colunas da matriz
range(m_bruta)
range(euclid)
min(euclid)
max(euclid)
mean(euclid) #CENTROIDE!! OU Grand mean
sd(euclid) #Standard deviation
centroide <- mean(euclid)
centroide
length(euclid)
m <- nrow(m_bruta)
m
m*(m-1)/2
summary(euclid)
Sumario1 <- cbind(min(euclid),
                  max(euclid),
                  sd(euclid),
                  mean(euclid),
                  length(euclid))
colnames(Sumario1) <- c("Minimo", "Maximo", "Desv.Padr", "Media", "m(m-1)/2")
rownames(Sumario1) <- ("Valores")
Sumario1
range(euclid)
par(mfrow=c(2,1))
hist(euclid,
```

```

breaks = 15, #determina o no. de colunas do histograma
xlim = range(floor(min(euclid)), ceiling(max(euclid))),
xlab = "Distr. de Frequências",
freq = FALSE)
curve(dnorm(x, mean=mean(euclid), sd=sd(euclid)), add=TRUE)
boxplot.default(euclid, horizontal = TRUE, frame = FALSE,
                xlab="Distr. de Frequências",
                ylim=c(floor(min(euclid)), ceiling(max(euclid)))) #Limites do eixo Y
## dev.off()
library(matrixStats)
library(moments)
euclid_ma <- as.matrix(euclid)
euclid_ma
range(euclid_ma) #valor errado
mean(euclid_ma) #valor errado
sd(euclid_ma) #valor errado
is.na(euclid_ma) <- euclid_ma==0 #atribui n.a. aos valores = 0
mean(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
mean(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
sd(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
#colMeans(euclid_ma, na.rm=T) #omite valores n.a. do calculo
#rowMeans(euclid_ma, na.rm=T) #omite valores n.a. do calculo
centroide_ma <- mean(euclid_ma, na.rm=T)
av.dist <- (as.matrix(colMeans(euclid_ma, na.rm=T)))
av.desvpad <- (as.matrix(colSds(euclid_ma, na.rm=T)))
dp.centroide_ma <- (av.dist-centroide_ma)/(colSds(av.dist)) #ou z-scores
list <- as.matrix(cbind(av.dist, av.desvpad, dp.centroide_ma))
list
colnames(list, do.NULL = FALSE)
colnames(list) <- c("Av.Dist", "Av.StDev", "DP.Centroide")
list2 <- list[order(list[,1], decreasing = TRUE),] #[,1] ou o nome da coluna
list2
par(mfrow=c(3,1))
hist(list2[, "Av.Dist"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias (em dp) para o centroide",
      main = "Distribuição de Frequência da distância média para o centroide",
      xlim = range(floor(min(av.dist)), ceiling(max(av.dist))), #substitua aqui o menor valor
      freq = T)
hist(list2[, "Av.Dist"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias (em dp) para o centroide",
      main = "Curva de normalidade ajustada para a Distribuição de Frequência",
      xlim = range(floor(min(av.dist)), ceiling(max(av.dist))), #substitua aqui o menor valor
      freq = F)

```

```

    freq = F)
curve(dnorm(x, mean=mean(list2[, "Av.Dist"]), sd=sd(list2[, "Av.Dist"])), add=TRUE)
boxplot.default(list2[, "Av.Dist"], horizontal = TRUE, frame = FALSE,
               xlab="Distr. de Frequências",
               ylim=c(floor(min(av.dist)), ceiling(max(av.dist)))) #substitua aqui o menor e maior
par(mfrow=c(1,1))
range(dp.centroide_ma)
par(mfrow=c(3,1))
hist(list2[, "DP.Centroide"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias dos desvios padões para o centroide",
      main = "Distribuição de Frequência dos desvio padões das distâncias médias para o centroide",
      xlim = range(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma))), #substitua aqui o menor e maior
      freq = T)
hist(list2[, "DP.Centroide"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias dos desvios padrões das distâncias médias para o centroide",
      main = "Curva de normalidade ajustada para a Distribuição de Frequência",
      xlim = range(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma))), #substitua aqui o menor e maior
      freq = F)
curve(dnorm(x, mean=mean(list2[, "DP.Centroide"]), sd=sd(list2[, "DP.Centroide"])), add=TRUE)
boxplot.default(list2[, "DP.Centroide"], horizontal = TRUE, frame = FALSE,
               xlab="Distr. de Frequências",
               ylim=c(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma)))) #substitua aqui o menor e maior
par(mfrow=c(1,1))
## dev.off()
cutoff <- 2.0
library(gt)
format(cutoff, nsmall = 1)
listf <- as.data.frame(list2)
listf$Outliers <- ifelse(listf$DP.Centroide>cutoff #CUTOFF MENOR QUE 'cutoff'
                           & listf$DP.Centroide<cutoff, #CUTOFF MAIOR QUE 'cutoff'
                           "", "OUT")
listf
gt(cbind(Sitios=rownames(listf),listf))
part <- c("B-A-MU4", "B-A-MU3", "S-A-TA4", "S-R-CT3")
part
m_bruta_part <- m_bruta[!(row.names(m_bruta) %in% c(part)),]
#m_bruta_part
sum <- colSums(m_bruta_part)
sum
zero_sum_cols <- names(which(colSums(m_bruta_part) == 0))
zero_sum_cols #nomes das espécies zeradas
m_bruta_part2 <- m_bruta_part[(colSums(m_bruta_part) != 0)] #em != a exclamação inverte o sentido

```

```

zero_sum_cols2 <- names(which(colSums(m_bruta_part2) == 0))
zero_sum_cols2 #nomes das espécies zeradas
sum<-colSums(m_bruta_part2)
sum
#m_bruta_part2
#m_bruta_part2 <- as.matrix(m_bruta_part2)
str(m_bruta_part2)
length(as.matrix(m_bruta_part2))
## df <- data.frame(Sites = rownames(m_bruta), m_bruta,
##                     row.names = NULL,
##                     check.names = FALSE) #add titulo a primeira coluna
##
## write.table(m_bruta_part2, "m_trabcsv.csv",
##             sep = ";", dec = ".",
##             row.names = TRUE,
##             quote = TRUE,
##             append = FALSE)
##
## m_trab <- read.csv("m_trabcsv.csv",
##                     sep = ";",
##                     row.names = 1,
##                     header = TRUE,
##                     na.strings = NA,
##                     check.names = FALSE, #impede que o R mude os nomes das colunas
##                     col.names = gsub("(^|_)([a-z])", "\\\1\\\\U\\\\2",
##                                     names(m_trab), perl = TRUE))

```

Referências

Chapter 16

R Modulo 10-1

Análise de Ordenação - NMDS

RESUMO

O Escalonamento Multidimensional Não-Métrico, também conhecido como NMDS ou MDS não-métrico (Multidimensional Scaling), é uma técnica estatística útil em situações onde as relações entre os pontos são importantes, mas as distâncias exatas entre eles podem não ser facilmente quantificáveis ou interpretáveis. Além da Ecologia, o NMDS é frequentemente usado em áreas como análise de dados, psicometria, ciência social, visualização de dados e aprendizado de máquina.

Apresentação

O Escalonamento Multidimensional Não-Métrico, também conhecido como NMDS ou MDS não-métrico (Multidimensional Scaling), é uma técnica estatística utilizada para representar dados com muitas dimensões em um espaço dimensional menor, de forma que a estrutura dos dados originais seja preservada o máximo possível.

Ao contrário do MDS métrico, que preserva as distâncias entre os pontos, o MDS não-métrico preserva apenas as relações de ordem ou similaridade entre os pontos, sem assumir uma relação linear com as distâncias reais. Em outras palavras, ele se concentra em preservar as relações de proximidade entre os pontos, mas não necessariamente suas distâncias exatas.

A técnica funciona construindo uma matriz de similaridade a partir dos dados originais e, em seguida, posicionando os pontos em um espaço de menor dimensão de forma que as distâncias ou similaridades relativas sejam mantidas o mais fiel possível à matriz original.

O MDS não-métrico é útil em situações onde as relações entre os pontos são importantes, mas as distâncias exatas entre eles podem não ser facilmente quantificáveis ou interpretáveis. Além da Ecologia, o NMDS é frequentemente usado em áreas como análise de dados, psicometria, ciência social, visualização de dados e aprendizado de máquina.

16.1 Algoritmo da NMDS

O algoritmo de uma NMDS (Non-Metric Multidimensional Scaling) funciona de maneira a posicionar os pontos de dados em um espaço de menor dimensão de forma que as relações de ordem ou similaridade entre os pontos sejam preservadas o máximo possível.

Aqui está uma visão geral de como o algoritmo de NMDS opera:

1. **Matriz de Similaridade:** O algoritmo começa com uma matriz de similaridade ou dissimilaridade que captura as relações entre os pontos de dados. Essa matriz pode ser baseada em diferentes medidas, como correlações, coeficientes de similaridade, distâncias euclidianas, entre outras.
2. **Inicialização:** Inicialmente, os pontos são atribuídos a posições aleatórias ou em uma configuração inicial no espaço de menor dimensão.
3. **Ajuste Iterativo:** O algoritmo itera para melhorar as posições dos pontos no espaço de menor dimensão. Durante cada iteração, os pontos são movidos em direção a uma configuração que melhor preserve as relações de similaridade ou dissimilaridade conforme definido pela matriz de similaridade.
4. **Critério de Convergência:** O processo iterativo continua até que um critério de convergência seja alcançado. Isso pode ser baseado na estabilidade das posições dos pontos ou na convergência de uma função de custo que quantifica a discrepância entre as relações de similaridade na configuração original e na configuração reduzida.
5. **Saída:** Uma vez que o algoritmo convergiu, as posições finais dos pontos no espaço de menor dimensão são utilizadas como representações dos dados para visualização ou análise.

É importante notar que, embora o NMDS preserve as relações de ordem ou similaridade entre os pontos, não há garantia de que as distâncias absolutas entre os pontos no espaço de menor dimensão correspondam às distâncias reais entre os pontos nos dados originais. O objetivo principal é preservar a estrutura relativa dos dados em um espaço de menor dimensão, facilitando a interpretação e visualização dos padrões nos dados.

16.2 Organização básica

Apagamos os gráficos, se houver algum, limpamos a memória e o console.

```
dev.off()
rm(list=ls(all=TRUE))
cat("\014")
#apaga os graficos, se houver algum, limpa a memoria e o console
```

16.3 Pacotes do módulo

Instalando os pacotes necessários para esse módulo. Nos computadores do Laboratório de Ecologia não instale esses pacotes, eles já estão instalados.

```
install.packages("openxlsx")
install.packages("rgl")

#dir <- getwd() #criamos um vetor com o diretório de trabalho
#shell.exec(dir) #abre o diretório de trabalho no Windows Explorer
library(openxlsx)
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06p-peixes.xlsx",
                    rowNames = T, colNames = T,
                    sheet = "Sheet1")
str(ppbio)
#View(ppbio)
ppbio[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
```

16.4 REINÍCIO 1

Aqui substitui-se uma nova matriz de dados, caso seja necessário refazer a análise com uma matriz gerada nesse código.

```
m_trab <- (ppbio) # <1>
```

1. Substitua a nova matriz aqui. Caso seja necessário.

16.5 Classificação 1: Matriz comunitária

Para conhecemos os dados, vamos criar uma classificação baseada na distância Bray-Curtis e UPGMA como método de fusão, a partir das matrizes de dados de interesse `ppbio**` com suas devidas relativizações e transformações.

16.5.1 Dendrograma e Heatmap 1

```

#Dendrograma
library(vegan)
m_trns <- asin(sqrt(decostand(m_trab,
                                method="total", MARGIN = 2)))
#m_trns <- sqrt(m_trab)
vegdist <- vegdist(m_trns, method = "bray",
                     diag = TRUE,
                     upper = FALSE)
cluster_uas <- hclust(vegdist, method = "average")
plot (cluster_uas, main = "Cluster Dendrogram - Bray-Curtis da Matriz Comunitária",
      hang = 0.1) #testar com -0.1
#rect.hclust(cluster_uas, k = 3, h = NULL)
#h = 0.8 fornece os grupos formados na altura h
as.matrix(vegdist)[1:6, 1:6]

#Heatmap
library("gplots")
heatdist <- as.matrix(vegdist)
col <- rev(heat.colors(999)) #rev() reverte as cores do heatmap
heatmap.2(x=as.matrix(vegdist)), #objetos x objetos
          Rowv = as.dendrogram(cluster_uas),
          Colv = as.dendrogram(cluster_uas),
          key = T, tracecol = NA, revC = T,
          col = heat.colors, #dissimilaridade = 1 - similaridade
          density.info = "none",
          xlab = "UA's", ylab = "UA's",
          mar = c(6, 6) + 0.2)
cluster_spp <- hclust((vegdist(t(m_trns), method = "bray",
                                 diag = TRUE,
                                 upper = FALSE)), method = "average")
plot (cluster_spp, main = "Dendrograma dos atributos")
heatmap.2(t(as.matrix(m_trns)), #objetos x atributos
          Colv = as.dendrogram(cluster_uas),
          Rowv = as.dendrogram(cluster_spp),
          key = T, tracecol = NA, revC = T,
          col = col,
          density.info = "none",
          xlab = "Unidades amostrais", ylab = "Espécies",
          mar = c(6, 6) + 0.1) # adjust margin size

```

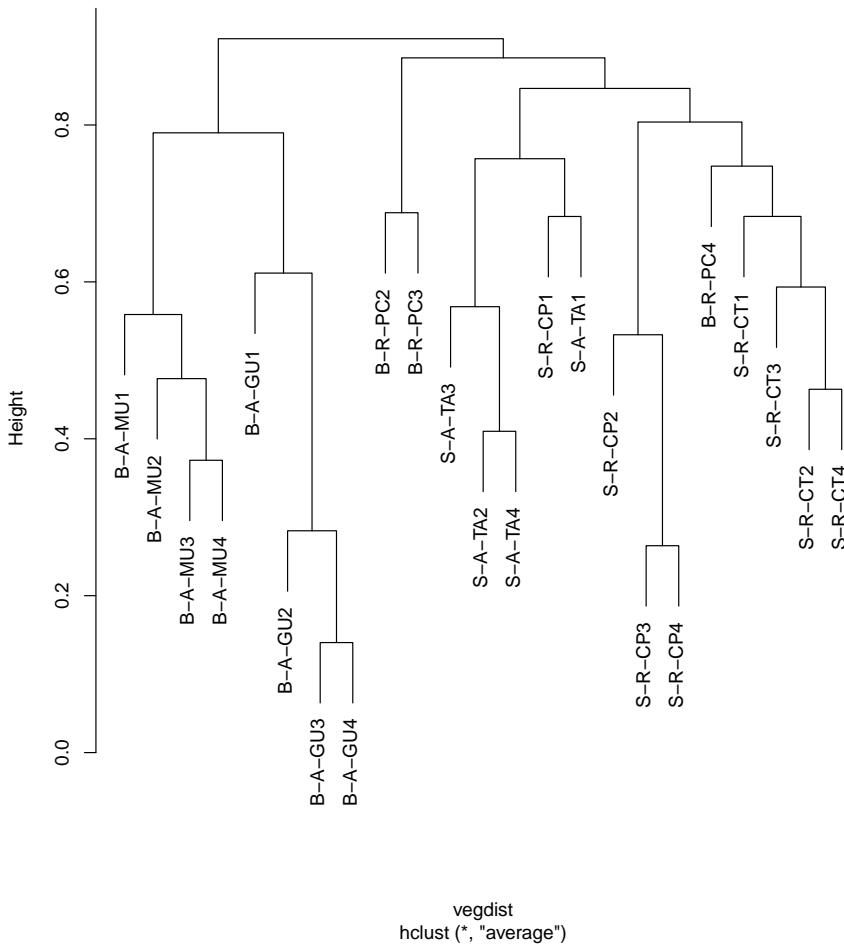
```

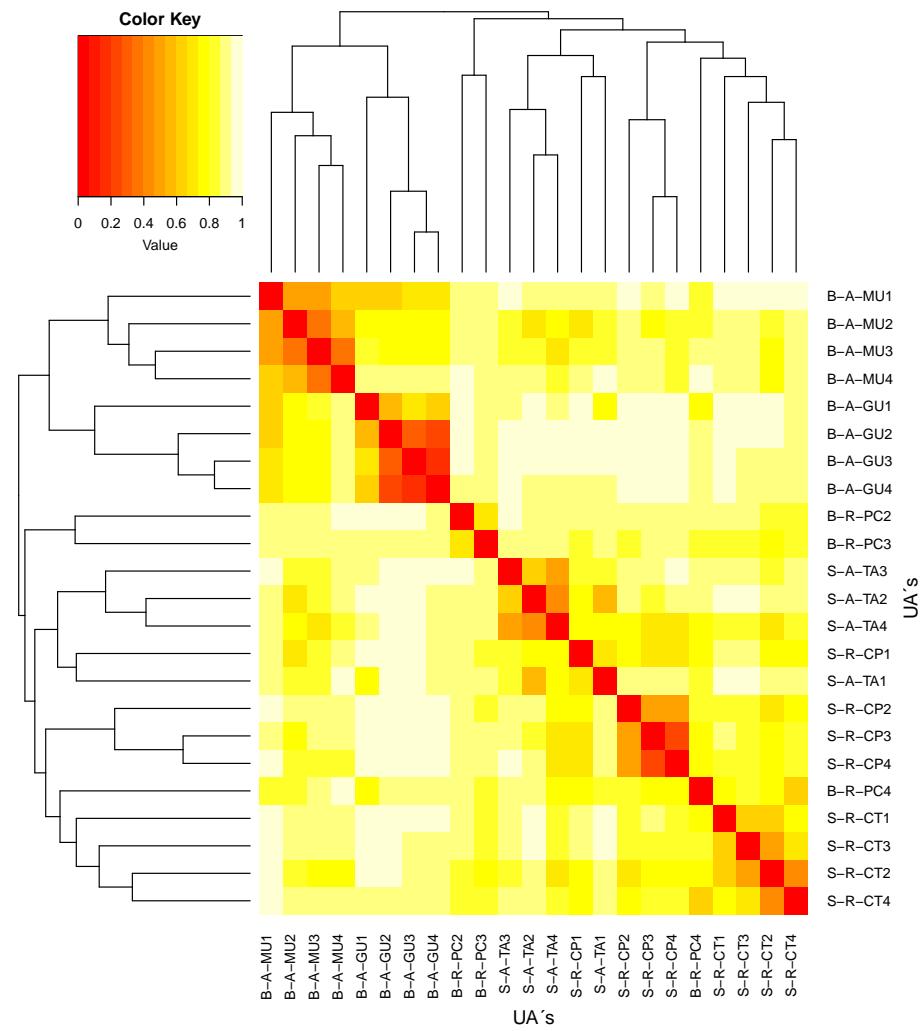
##           S-R-CT1   S-R-CP1   S-A-TA1   S-R-CT2   S-R-CP2   S-A-TA2
## S-R-CT1 0.0000000 0.8743721 0.9338269 0.6274997 0.8106894 0.9420728
## S-R-CP1 0.8743721 0.0000000 0.6833816 0.7759468 0.7726098 0.7342613

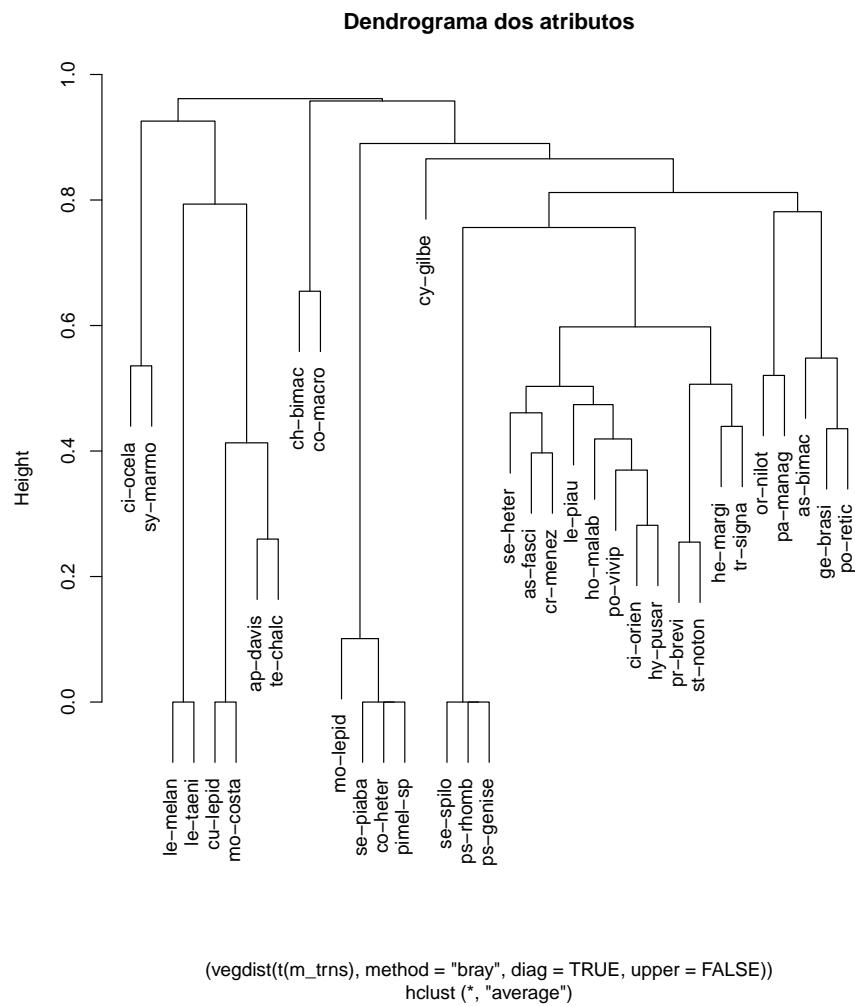
```

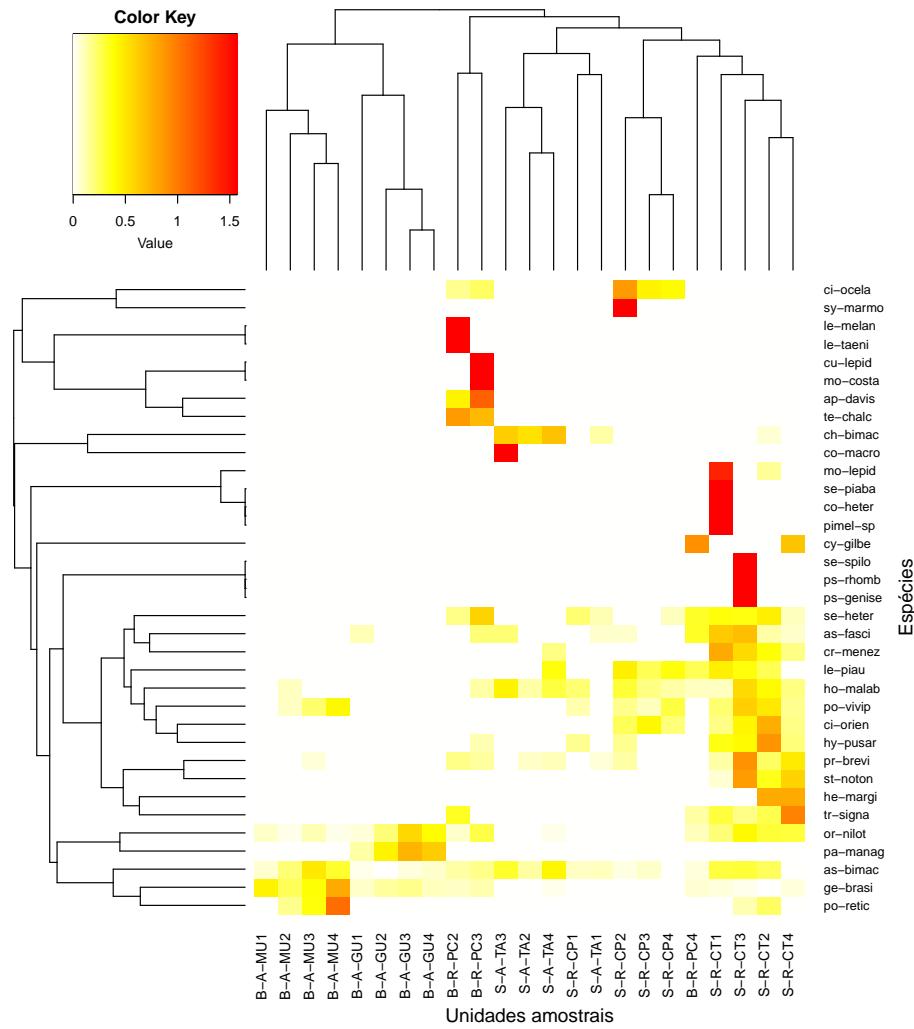
```
## S-A-TA1 0.9338269 0.6833816 0.0000000 0.8789631 0.9178304 0.5700984
## S-R-CT2 0.6274997 0.7759468 0.8789631 0.0000000 0.7280378 0.8836068
## S-R-CP2 0.8106894 0.7726098 0.9178304 0.7280378 0.0000000 0.8915271
## S-A-TA2 0.9420728 0.7342613 0.5700984 0.8836068 0.8915271 0.0000000
```

Cluster Dendrogram – Bray–Curtis da Matriz Comunitária









16.5.2 Histórico das fusões 1

Criamos agora o histórico das fusões dos objetos. Na tabela gerada, as duas primeiras colunas (No. e UA) representam o número (No.) atribuído a cada unidade amostral (UA). As duas colunas subsequentes (Cluster1 e Cluster2) representam o par de objetos (indicado pelo sinal de "-") ou grupo de objetos (indicado pela ausência do sinal de "-") que foram agrupadas. A coluna Height, indica o valor de similaridade na qual um dado par de objetos (ou grupo de objetos) foi agrupado. O valor aproximado de Height também pode ser visualizado no eixo do dendrograma. Por último, na coluna Histórico, é mostrada a sequência das fusões da primeira até a $m-1$ última fusão entre os dois últimos grupos. Nesse caso, 22.

```

library(tidyverse)
library(gt)
merge <- as.data.frame(cluster_uas$merge)
merge[nrow(merge)+1,] = c("0","0")
height <- as.data.frame(round(cluster_uas$height, 2))
height[nrow(height)+1,] = c("1.0")
fusoes <- data.frame(Cluster_uas = merge, Height = height)
colnames(fusoes) <- c("Cluster1", "Cluster2", "Height")
UA <- rownames_to_column(as.data.frame(m_trns[, 0]))
colnames(UA) <- c("UAs")
No.UA <- 1:nrow(fusoes)
fusoes <- cbind(No.UA, UA, fusoes)
fusoes$Histórico <- 1:nrow(fusoes)
#fusoes
gt(fusoes)

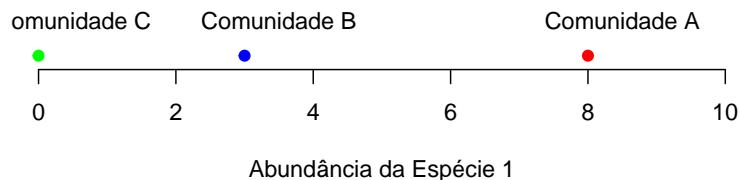
```

No.UA	UAs	Cluster1	Cluster2	Height	Histórico
1	S-R-CT1	-20	-23	0.14	1
2	S-R-CP1	-8	-11	0.26	2
3	S-A-TA1	-17	1	0.28	3
4	S-R-CT2	-19	-22	0.37	4
5	S-R-CP2	-6	-12	0.41	5
6	S-A-TA2	-4	-10	0.46	6
7	S-R-CT3	-16	4	0.48	7
8	S-R-CP3	-5	2	0.53	8
9	S-A-TA3	-13	7	0.56	9
10	S-R-CT4	-9	5	0.57	10
11	S-R-CP4	-7	6	0.59	11
12	S-A-TA4	-14	3	0.61	12
13	B-A-MU1	-2	-3	0.68	13
14	B-A-GU1	-1	11	0.68	14
15	B-R-PC2	-15	-18	0.69	15
16	B-A-MU2	-21	14	0.75	16
17	B-A-GU2	10	13	0.76	17
18	B-R-PC3	9	12	0.79	18
19	B-A-MU3	8	16	0.8	19
20	B-A-GU3	17	19	0.85	20
21	B-R-PC4	15	20	0.89	21
22	B-A-MU4	18	21	0.91	22
23	B-A-GU4	0	0	1.0	23

16.6 Ordenação

O Escalonamento Multidimensional Não-Métrico (NMS ou NMDS) é uma ferramenta comumente usada para examinar a composição da comunidade. Vamos estabelecer algumas bases conceituais. Considere um único eixo de abundância representando uma única espécie Fonte:

```
plot(0:10,0:10,type="n",axes=F,xlab="Abundância da Espécie 1",ylab="")
axis(1)
points(8,0, pch=19,col="red"); text(8.5,1,labels="Comunidade A")
points(3,0, pch=19,col="blue"); text(3.5,1,labels="Comunidade B")
points(0,0, pch=19,col="green" ); text(0.5,1,labels="Comunidade C")
```



Podemos plotar cada comunidade nesse eixo dependendo da abundância da espécie 1 dentro dela.

```
cat('
|-----|-----|
| Comunidade | Sp1 |
|-----|-----|
| A (verm)   |  8  |
| B (azul)    |  3  |
| C (verde)   |  0  |
|-----|-----|
')
```

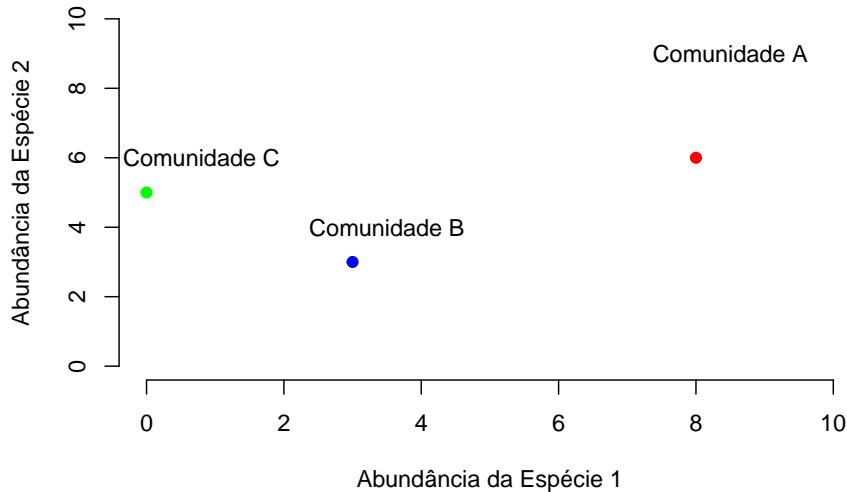
```
##  
## |-----|----|  
## | Comunidade | Sp1 |  
## |-----|----|  
## | A (verm) | 8 |  
## | B (azul) | 3 |  
## | C (verde) | 0 |  
## |-----|----|
```

Agora, considere um segundo eixo de abundância representando uma espécie diferente. As comunidades podem ser plotadas ao longo de ambos os eixos dependendo da abundância das espécies dentro delas.

```
cat('  
|-----|----|----|  
| Comunidade | Sp1 | Sp2 |  
|-----|----|----|  
| A (verm) | 8 | 6 |  
| B (azul) | 3 | 3 |  
| C (verde) | 0 | 5 |  
|-----|----|----|  
)
```

|-----|—|—| | Comunidade | Sp1 | Sp2 | |-----|—|—| | A (verm) |
8 | 6 | | B (azul) | 3 | 3 | | C (verde) | 0 | 5 | |-----|—|—|

```
plot(0:10,0:10,type="n",xlab="Abundância da Espécie 1",  
     ylab="Abundância da Espécie 2", frame.plot = F)  
points(8,6, pch = 19, col = "red"); text(8.5,9,labels="Comunidade A")  
points(3,3, pch = 19, col = "blue"); text(3.5,4,labels="Comunidade B")  
points(0,5, pch = 19, col = "green"); text(0.8,6,labels="Comunidade C")
```



Agora, considere um TERCEIRO eixo de abundância representando mais uma espécie.

```
cat('
|-----|-----|-----|-----|
| Comunidade | Sp1 | Sp2 | Sp3 |
|-----|-----|-----|-----|
| A (verm) | 8   | 6   | 7   |
| B (azul)  | 3   | 3   | 3   |
| C (verde) | 0   | 5   | 5   |
|-----|-----|-----|-----|
')
```

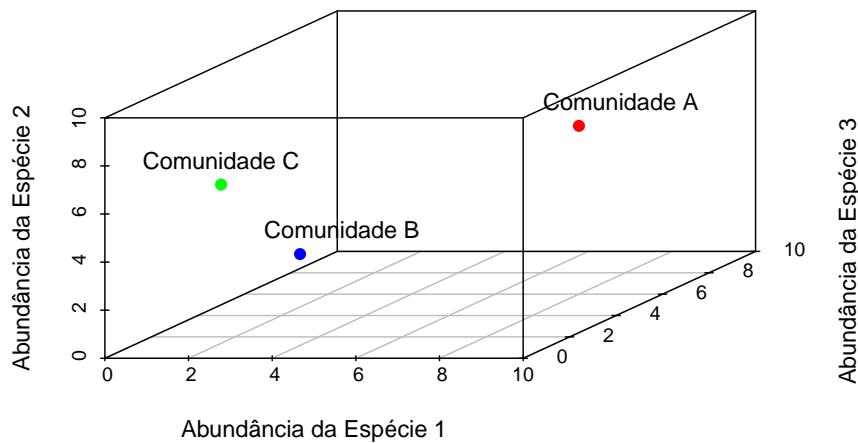
	Comunidade			Sp1	Sp2	Sp3					
A (verm)	8	6	7	B (azul)	3	3	3	C (verde)	0	5	5

Para isso, precisaremos carregar outro pacote. Para instalar o pacote, utiliza-se o seguinte comando:

```
install.packages("scatterplot3d")
```

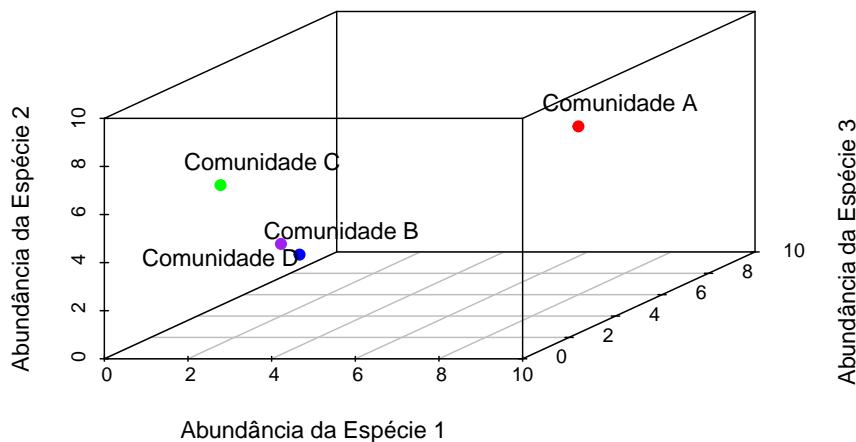
Este comando instalará o pacote “scatterplot3d” em seu ambiente R, permitindo que você crie gráficos tridimensionais para explorar a abundância de múltiplas espécies em suas comunidades.

```
library(scatterplot3d)
d=scatterplot3d(0:10,0:10,0:10,type="n",
               xlab="Abundância da Espécie 1",
               ylab="Abundância da Espécie 3",
               zlab="Abundância da Espécie 2",
               angle = 40)
d$points3d(8,6,7, pch = 19, col = "red"); text(d$xyz.convert(9,6,8),labels="Comunidade A")
d$points3d(3,3,3, pch = 19, col = "blue"); text(d$xyz.convert(4,3,4),labels="Comunidade B")
d$points3d(0,5,5, pch = 19, col = "green"); text(d$xyz.convert(0,5,6),labels="Comunidade C")
```



Agora, em uma matriz n x m,

```
cat('
-----|-----|-----|-----|-----|-----|-----|
| Comunidade | Sp1 | Sp2 | Sp3 | Sp4 | Sp5 |   m   |
|-----|-----|-----|-----|-----|-----|-----|
| A (azul)  |  8  |  6  |  7  |  3  |  5  |   .   |
| B (verde) |  3  |  3  |  3  |  2  |  1  |   .   |
| C (verm)  |  0  |  5  |  5  |  1  |  0  |   .   |
| D (roxo)  |  2  |  4  |  3  |  0  |  1  |   .   |
| n          |  .  |  .  |  .  |  .  |  .  |   .   |
|-----|-----|-----|-----|-----|-----|-----|
')
```

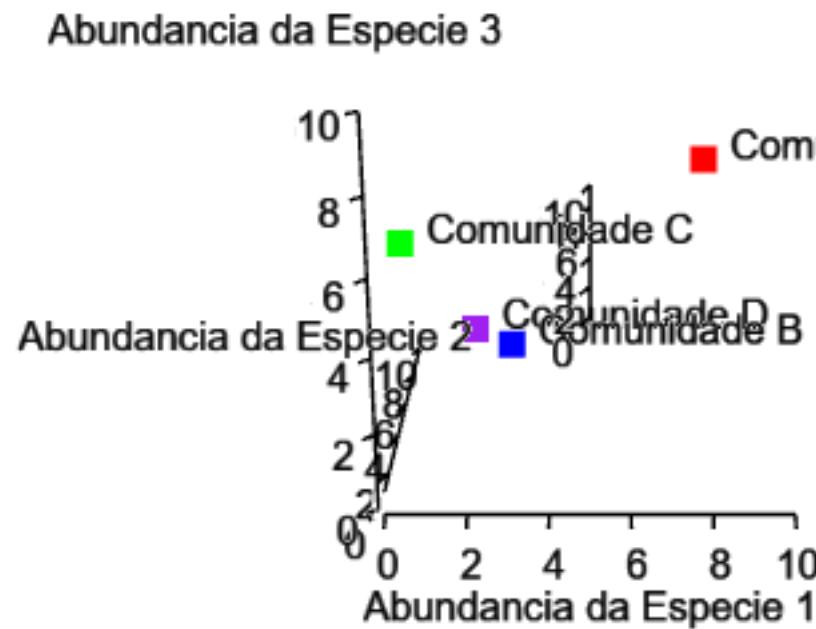


Podemos tentar um plot em 3D...

```
# Plot the 3D plot
# Open a new rgl device
open3d()

## wgl
## 1

# Add points and labels to the 3D plot
material3d(size = 10, id = NULL)
points3d(8, 6, 7, pch = 19, col = "red")
text3d(8, 6, 7, "Comunidade A", adj = c(-0.1, 0))
points3d(3, 3, 3, pch = 19, col = "blue")
text3d(3, 3, 3, "Comunidade B", adj = c(-0.1, 0))
points3d(0, 5, 5, pch = 19, col = "green")
text3d(0, 5, 5, "Comunidade C", adj = c(-0.1, 0))
points3d(10,10,10, pch = 19, col = "white")
points3d(0,0,0, pch = 19, col = "white")
points3d(2,4,3, pch = 19, col = "purple")
text3d(2,4,3, "Comunidade D", adj = c(-0.1, 0))
# Add axes
#?axes3d
#axes3d()
# Put 4 x-axes on the plot
#axes3d(c('x--', 'x-+', 'x+-', 'x++'))
## Use fixed axes
axes3d(c('x', 'y', 'z'))
axis3d('y', pos = c(5,NA,5))
title3d('', '', 'Abundancia da Especie 1', 'Abundancia da Especie 3', 'Abundancia da Especie 2')
# Rotate the plot interactively
rglwidget()
```



```
x <- c(0, 10, 0, 0)
y <- c(0, 0, 10, 0)
z <- c(0, 0, 0, 10)
labels <- c("Origin", "Sp1", "Sp3", "Sp2")
i <- c(1, 2, 1, 3, 1, 4)
A <- c(8,6,7)
B <- c(3,3,3)
C <- c(0,5,5)
##3d interface
open3d()
```

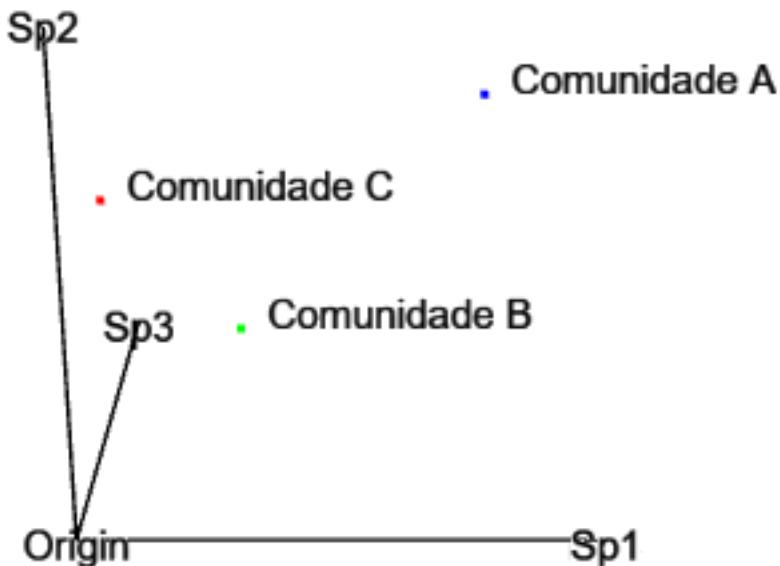
```
## wgl
##    5

text3d(x,y,z,labels)
points3d(A,col = 'blue')
points3d(B,col = 'green')
```

```

points3d(C,col = 'red')
text3d(8,6,7, "Comunidade A", adj = c(-0.1, 0))
text3d(3,3,3, "Comunidade B", adj = c(-0.1, 0))
text3d(0,5,5, "Comunidade C", adj = c(-0.1, 0))
segments3d(x[i], y[i], z[i])
rglwidget()

```



considere tantos eixos quanto espécies existentes em uma comunidade. Obviamente, não podemos visualizar isso além de 3 dimensões.

O objetivo do NMDS é representar a posição original das comunidades em um espaço multidimensional da forma mais precisa possível, usando um número reduzido de dimensões que podem ser facilmente plotadas e visualizadas.

Por ser uma ordenação **não-métrica** o NMDS não utiliza as abundâncias absolutas das espécies nas comunidades, mas sim a ordem do seu “rank” estatístico (ou sua “ordenação”). O uso de “ranks” ou ordenações omite alguns dos problemas associados ao uso de distâncias absolutas (por exemplo, sensibilidade à

transformação) e, como resultado, é uma técnica muito mais flexível, que aceita uma variedade de tipos de dados.

O procedimento NMDS segue um algorítimo iterativo que ocorre em várias etapas:

- (1) Definir as posições originais das comunidades em espaço multidimensional
- (2) Especificar o número m de dimensões reduzidas (tipicamente 2)
- (3) Construir uma configuração inicial das amostras em 2 dimensões
- (4) Regressar distâncias nesta configuração inicial contra as distâncias observadas (medidas)
- (5) Determinar o stress (discordância entre a configuração 2-D e os valores previstos da regressão). Se a configuração 2-D preservar perfeitamente os ranks originais de classificação, então um gráfico de um vs. o outro deve ser monotonicamente crescente (gráfico de Shepard). A extensão com que os pontos na configuração 2-D diferem dessa linha monotonicamente crescente determina o grau de stress
- (6) Se o stress for alto, reposicione os pontos em m dimensões na direção do stress decrescente e repita até que o stress esteja abaixo de um certo limite.

Geralmente, $\text{stress} < 0.05$ fornece uma excelente representação em dimensões reduzidas, < 0.1 é ótimo, < 0.2 é bom, e $\text{stress} > 0.3$ fornece uma representação ruim.

NOTA: A configuração final pode diferir dependendo da configuração inicial (que normalmente é aleatória) e do número de iterações do algorítmo, então é aconselhável executar o NMDS várias vezes e comparar a interpretação das soluções de menor stress.

Para começar, o NMDS requer uma matriz de distâncias (ou uma matriz de dissimilaridades). Distâncias Euclidianas brutas não são ideais para este propósito: elas são sensíveis às abundâncias totais, então podem tratar locais com um número semelhante de espécies como mais semelhantes, mesmo que as identidades das espécies sejam diferentes. Elas também são sensíveis às ausências de espécies, então podem tratar locais com o mesmo número de espécies ausentes como mais semelhantes.

Consequentemente, os ecologistas usam o cálculo de dissimilaridade de Bray-Curtis, que possui muitas propriedades ideais:

- É invariante a mudanças em unidades
- Não é afetado por adições/remoções de espécies que não estão presentes em duas comunidades
- Não é afetado pela adição de uma nova comunidade
- Pode reconhecer diferenças nas abundâncias totais quando as abundâncias relativas são iguais.

Para executar a NMDS, usa-se a função `metaMDS` do pacote `vegan`.

```
library(vegan)
```

`metaMDS` requer uma matriz de comunidade-por-espécies (CBE, “community-by-species”). De momento, cria-se essa matriz com dados amostrados aleatoriamente de um conjunto arbitrário de valores.

```
set.seed(2)
community_matrix=matrix(sample(1:100,300,replace=T),nrow=10,
                        dimnames=list(paste("community",1:10,sep=""),
                                      paste("sp",1:30,sep=""))))
```

A função `metaMDS` vai calcular as distâncias, executar o algoritimo iterativo, determina o stress ou fitting entre os conjuntos de dados e etc. Precisa-se apenas definir a matriz de comunidade-por-espécies. Nesse caso:

```
example_NMDS=metaMDS(community_matrix, #matriz CBS, definida anteriormente
                      k=2) #no. de redução de dimensões
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1486475
## Run 1 stress 0.1913689
## Run 2 stress 0.1908303
## Run 3 stress 0.1908305
## Run 4 stress 0.1849153
## Run 5 stress 0.1906937
## Run 6 stress 0.1486477
## ... Procrustes: rmse 0.0002739181 max resid 0.000426955
## ... Similar to previous best
## Run 7 stress 0.1849759
## Run 8 stress 0.1688829
## Run 9 stress 0.1650751
## Run 10 stress 0.1486475
## ... Procrustes: rmse 8.715875e-05 max resid 0.0001400462
## ... Similar to previous best
## Run 11 stress 0.164544
## Run 12 stress 0.3153591
## Run 13 stress 0.2198889
## Run 14 stress 0.2237308
## Run 15 stress 0.1699872
## Run 16 stress 0.1650804
## Run 17 stress 0.1906937
## Run 18 stress 0.1906937
## Run 19 stress 0.1494912
```

```
## Run 20 stress 0.1650751
## *** Best solution repeated 2 times
```

Deve-se observar cada iteração do NMDS até que uma solução seja alcançada (ou seja, o stress foi minimizado após algum número de reconfigurações dos pontos em 2 dimensões). Pode-se aumentar o número padrão de iterações usando o argumento `trymax` = . Isso pode ajudar a aliviar problemas de não convergência. Se o stress for alto, pode-se aumentar o número de dimensões para `k` = 3.

```
example_NMDS=metaMDS(community_matrix,k=2,trymax=100)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1486475
## Run 1 stress 0.1906937
## Run 2 stress 0.1486475
## ... New best solution
## ... Procrustes: rmse 4.504062e-05 max resid 7.46737e-05
## ... Similar to previous best
## Run 3 stress 0.164544
## Run 4 stress 0.2043556
## Run 5 stress 0.1486475
## ... New best solution
## ... Procrustes: rmse 3.786411e-05 max resid 5.798541e-05
## ... Similar to previous best
## Run 6 stress 0.1686179
## Run 7 stress 0.1486487
## ... Procrustes: rmse 0.001296765 max resid 0.002408379
## ... Similar to previous best
## Run 8 stress 0.1881025
## Run 9 stress 0.1751078
## Run 10 stress 0.1650804
## Run 11 stress 0.1486475
## ... Procrustes: rmse 0.0001191068 max resid 0.0001867147
## ... Similar to previous best
## Run 12 stress 0.1663641
## Run 13 stress 0.1744082
## Run 14 stress 0.1717124
## Run 15 stress 0.1751081
## Run 16 stress 0.1769791
## Run 17 stress 0.1906937
## Run 18 stress 0.1974143
## Run 19 stress 0.1650751
## Run 20 stress 0.1486475
## ... Procrustes: rmse 0.0001122687 max resid 0.0001737893
## ... Similar to previous best
```

```
## *** Best solution repeated 4 times
```

Pode-se agora examinar o objeto NMDS.

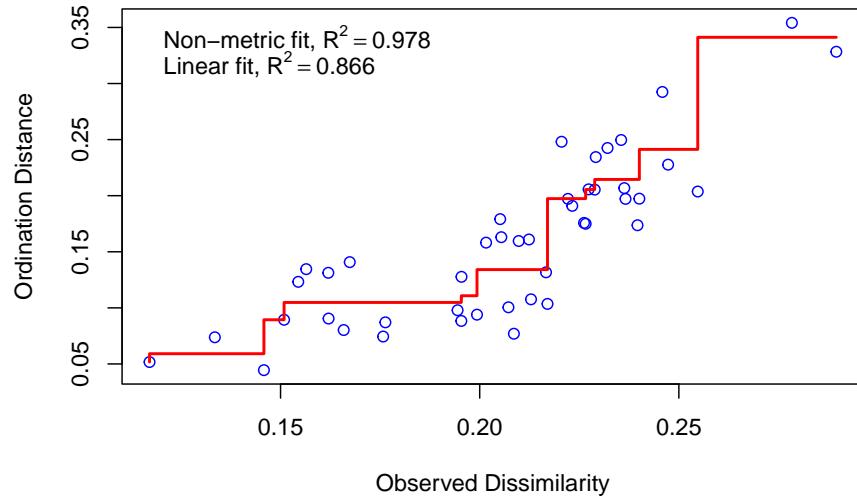
```
example_NMDS
```

```
##  
## Call:  
## metaMDS(comm = community_matrix, k = 2, trymax = 100)  
##  
## global Multidimensional Scaling using monoMDS  
##  
## Data:      wisconsin(sqrt(community_matrix))  
## Distance: bray  
##  
## Dimensions: 2  
## Stress:     0.1486475  
## Stress type 1, weak ties  
## Best solution was repeated 4 times in 20 tries  
## The best solution was from try 5 (random start)  
## Scaling: centring, PC rotation, halfchange scaling  
## Species: expanded scores based on 'wisconsin(sqrt(community_matrix))'
```

Nesse exemplo, `metaMDS` aplicou automaticamente uma transformação da raiz quadrada e calculou as distâncias de Bray-Curtis para a matriz de comunidade-por-local.

É necessário examinar o **Gráfico de Shepard**, que mostra a dispersão em torno da regressão entre as distâncias entre pontos da configuração final (distâncias entre cada par de comunidades) *versus* as dissimilaridades originais.

```
stressplot(example_NMDS)
```



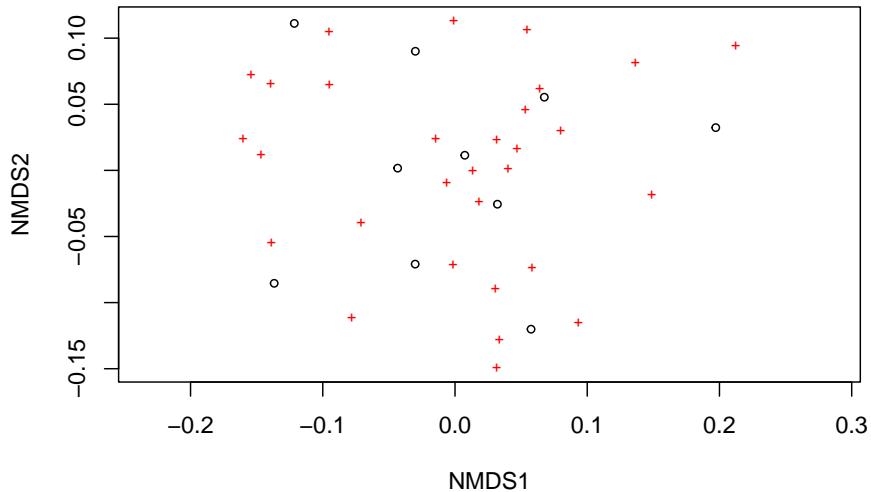
Uma dispersão grande ao redor da linha sugere que as dissimilaridades originais não são bem preservadas no número reduzido de dimensões.

Agora podemos plotar a NMDS

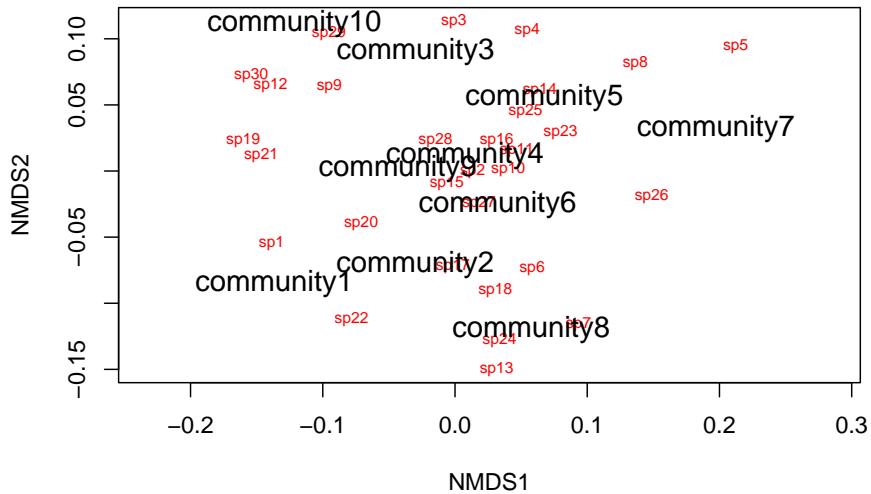
A ordenação mostra tanto as comunidades (“locais”, círculos abertos) quanto as espécies (cruzes vermelhas).

Aqui, pode-se usar as funções `ordiplot` e `orditorp` para adicionar texto ao gráfico no lugar dos pontos.

```
plot(example_NMDS)
```



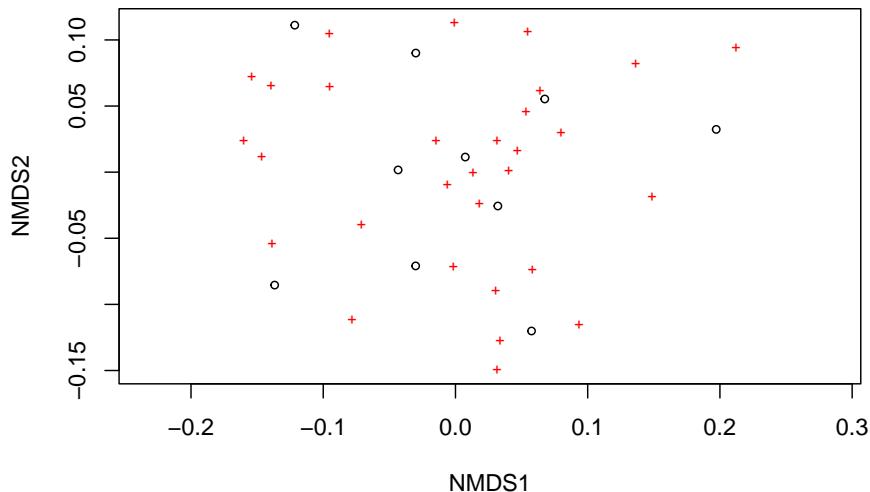
```
ordiplot(example_NMDS,type="n")
orditorp(example_NMDS,display="species",col="red",air=0.01)
orditorp(example_NMDS,display="sites",cex=1.25,air=0.01)
```



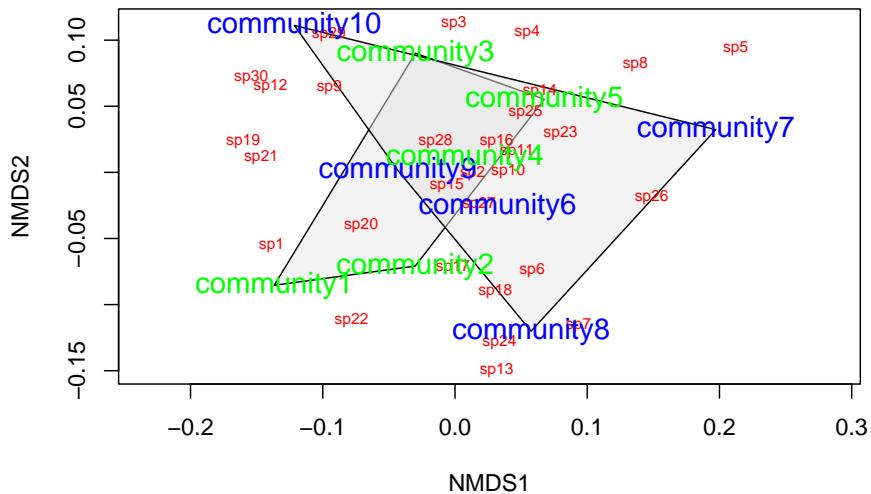
Existem algumas funções adicionais que podem ser de interesse. Pode-se desenhar polígonos convexos conectando os vértices dos pontos feitos por certos grupos de comunidades no gráfico.

Para isso, cria-se um vetor de valores de tratamento.

```
plot(example_NMDS)
```



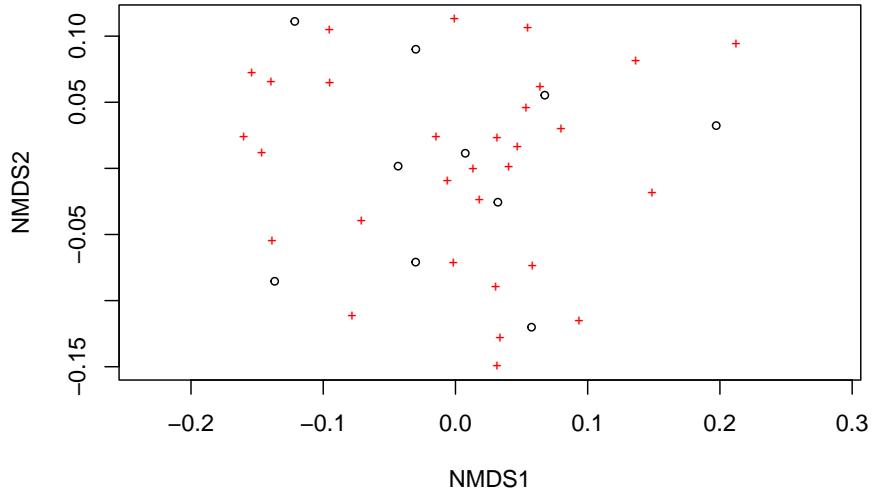
```
treat=c(rep("Treatment1",5),rep("Treatment2",5))
ordiplot(example_NMDS,type="n")
ordihull(example_NMDS,groups=treat,draw="polygon",col="grey90",
          label=FALSE)
orditorp(example_NMDS,display="species",col="red",air=0.01)
orditorp(example_NMDS,display="sites",col=c(rep("green",5),rep("blue",5)),
          air=0.01,cex=1.25)
```



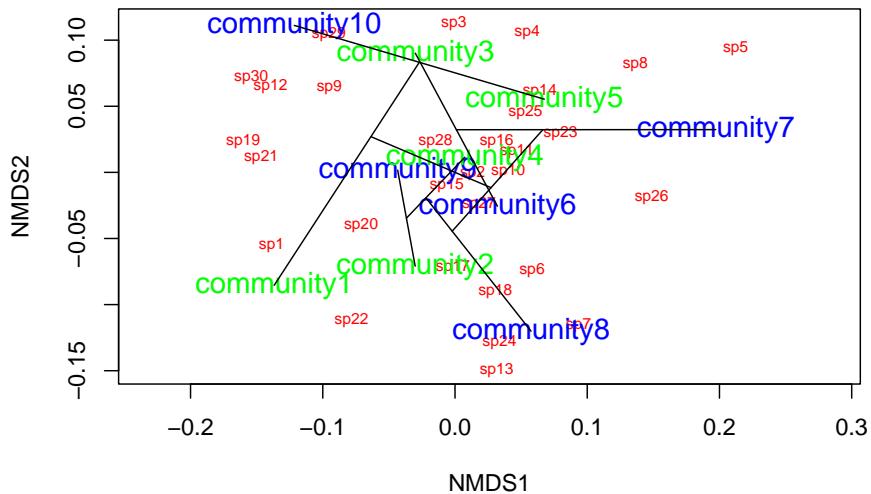
Essa é uma maneira intuitiva de entender como as comunidades e espécies se agrupam com base nos tratamentos possíveis. Nesse caso, comunidades 1-5 em verde e comunidades 6-10 em azul. Também é possível plotar elipses e “gráficos de aranha” usando as funções `ordielipse` e `orderspider`, que enfatizam o centróide das comunidades em cada tratamento.

Outra alternativa é plotar uma “árvore de abrangência mínima” (da função `hclust`), que agrupa as comunidades com base em suas dissimilaridades originais e projeta o dendrograma no gráfico 2D.

```
plot(example_NMDS)
```



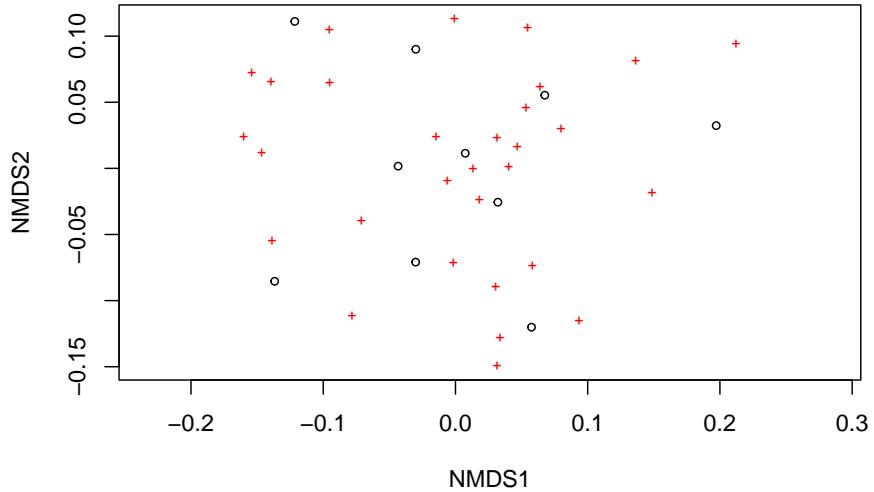
```
ordiplot(example_NMDS,type="n")
orditorp(example_NMDS,display="species",col="red",air=0.01)
orditorp(example_NMDS,display="sites",col=c(rep("green",5),rep("blue",5)),
         air=0.01,cex=1.25)
ordicluster(example_NMDS,hclust(vegdist(community_matrix,"bray")))
```



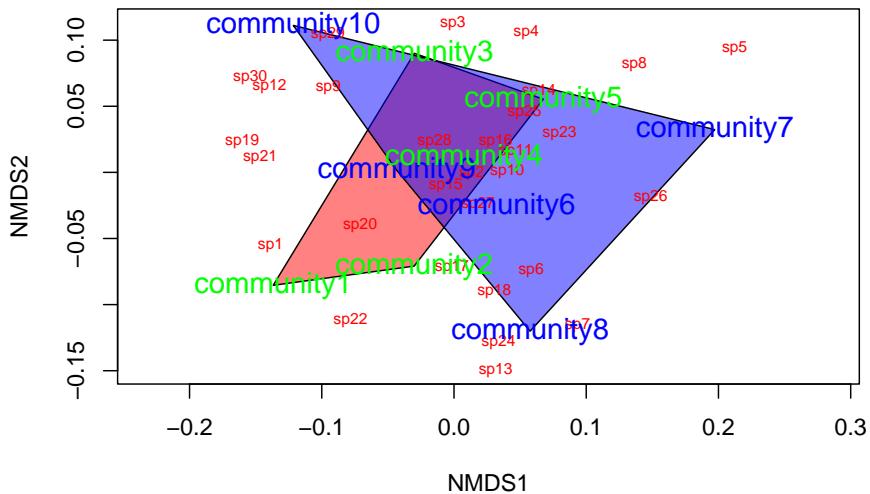
Observe que o agrupamento é baseado em distâncias de Bray-Curtis. Este é um método sugerido para verificar a precisão do gráfico 2D.

Pode-se ainda plotar os polígonos convexos, elipses, gráficos de aranha, etc., coloridos com base nos tratamentos. Mas primeiro, cria-se um vetor de valores de cor com a mesma extensão do vetor de valores de tratamento.

```
plot(example_NMDS)
```



```
colors=c(rep("red",5),rep("blue",5))
ordiplot(example_NMDS,type="n")
#Plot convex hulls with colors baesd on treatment
for(i in unique(treat)) {
  ordihull(example_NMDS$point[grep(i,treat),],draw="polygon",groups=treat[treat==i],co
  orditorp(example_NMDS,display="species",col="red",air=0.01)
  orditorp(example_NMDS,display="sites",col=c(rep("green",5),rep("blue",5)),
            air=0.01,cex=1.25)
```

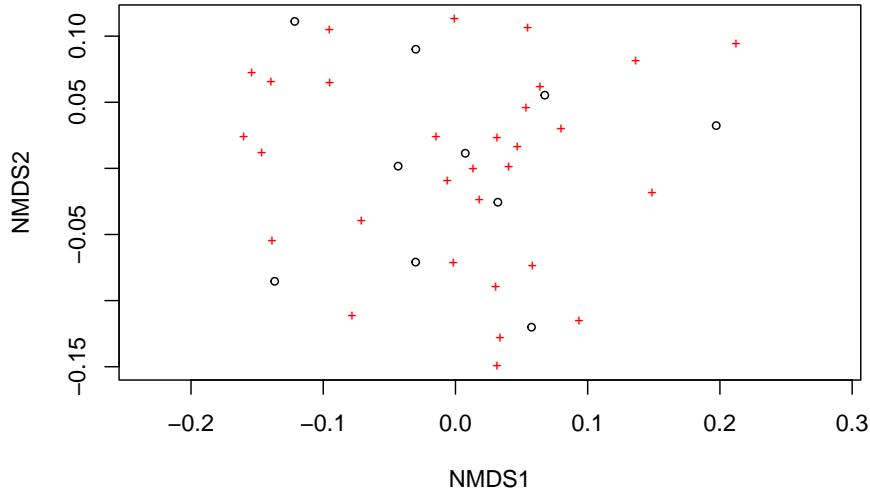


Esse agrupamento é baseado em distâncias de Bray-Curtis. Este é um método sugerido para verificar a precisão do gráfico 2D.

Pode-se plotar os polígonos convexas, elipses, gráficos de aranha, etc., coloridos com base nos tratamentos. Primeiro, cria-se um vetor de valores de cor com a mesma extensão do vetor de valores de tratamento.

Se o tratamento for uma variável contínua, deve-se considerar linhas de mapeamento de contorno sobrepostas no gráfico. Para este exemplo, considera-se que os tratamentos foram aplicados ao longo de um gradiente de elevação. Podemos definir elevações aleatórias para o mesmo exemplo.

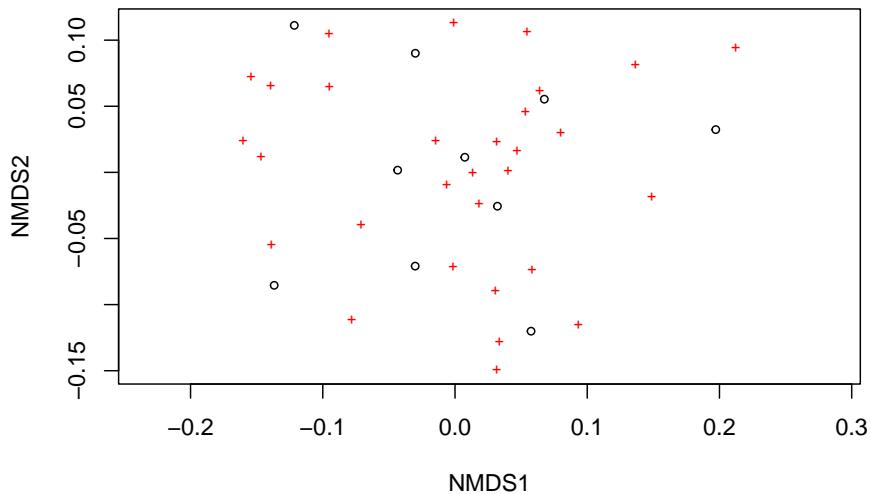
```
plot(example_NMDS)
```



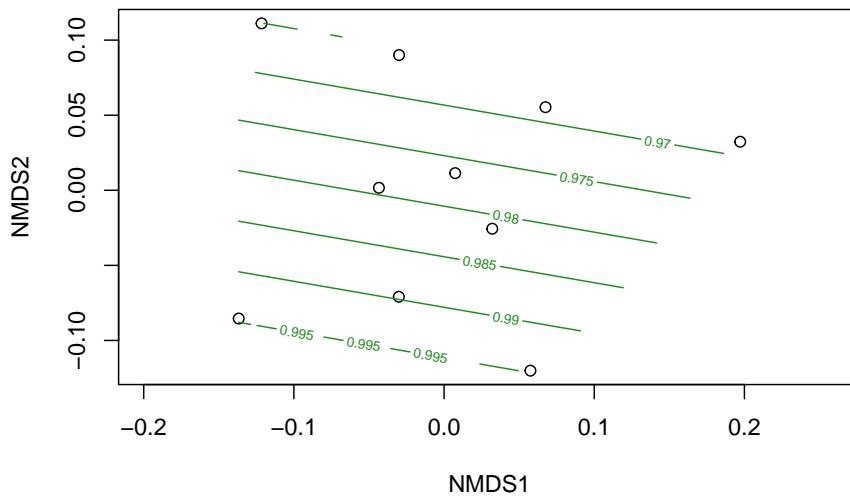
```
elevation=runif(10,0.5,1.5)
```

Agora usa-se a função `ordisurf` para plotar as linhas de contorno. A quantidade de linhas de contorno é definida pelo comando `knots` = . Se `knots` = 0 ou `knots` = 1 a função `ordisurf` ajustará uma tendência linear a superfície, e se `knots` = 2 a função ajustará uma tendência quadrática, ao invés de linhas retas.

```
plot(example_NMDS)
```



```
ordisurf(example_NMDS,elevation,main="",col="forestgreen")
```



```
##  
## Family: gaussian
```

```

## Link function: identity
##
## Formula:
## y ~ s(x1, x2, k = 10, bs = "tp", fx = FALSE)
##
## Estimated degrees of freedom:
## 0.189  total = 1.19
##
## REML score: 1.428577

```

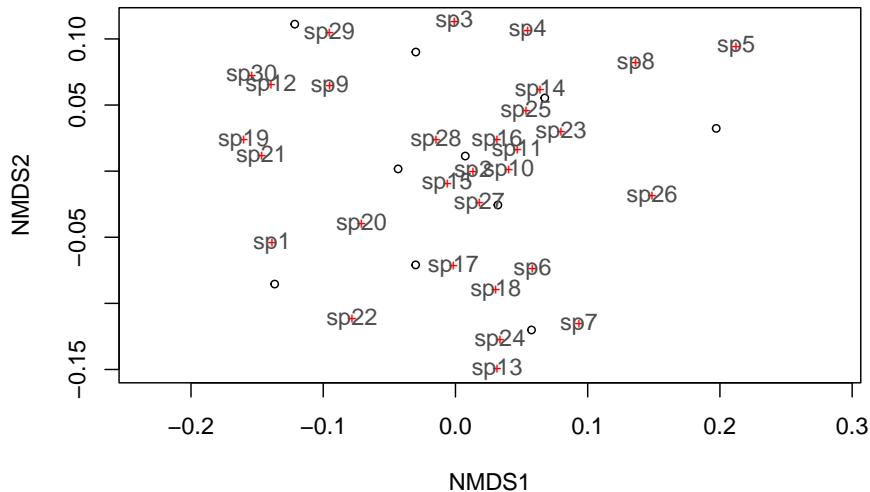
Por fim, exibe-se as espécies no gráfico.

```

plot(example_NMDS)

orditorp(example_NMDS,display="species",col="grey30",air=0.1,cex=1)

```



Refaça a análise com os dados fornecidos.

Apêndices

Sites consultados

<https://youtu.be/OMrtxobDhrM> <https://youtu.be/zjxE2KwNO0M> <https://youtu.be/Kl49qI3XJKY> <https://youtu.be/paO9-PLXwD0> <https://youtu.be/QljEeBei-JA> <https://youtu.be/4tQJI-g9wao> <https://youtu.be/M-Ing955VEI>

Referências

Part III

**ANÁLISES
MULTIVARIADAS**

Chapter 17

R Módulo 7 - Análise de Classificação Hierárquica - SAHN

RESUMO

Análise de Classificação Hierárquica (SAHN) é um método de análise de dados utilizado para identificar grupos ou clusters de objetos com base em sua similaridade. É amplamente utilizada em várias áreas, incluindo bioinformática, análise de dados espaciais, agronomia, entre outros. A análise de classificação hierárquica é uma técnica exploratória importante que pode ajudar a identificar padrões e relações em dados não estruturados. No entanto, é importante lembrar que a interpretação dos resultados pode ser subjetiva e depender do contexto da aplicação.

Apresentação

Análise de Classificação Hierárquica (do inglês, “Hierarchical Agglomerative Clustering Analysis”, ou simplesmente “SAHN”) é um método de análise de dados utilizado para identificar grupos ou “clusters” de objetos com base em sua similaridade. Nessa análise, os objetos são inicialmente considerados como clusters individuais e, em seguida, os objetos mais similares são agrupados em um cluster maior, e assim por diante, até que todos os objetos estejam em um único cluster. Isso resulta em uma árvore hierárquica, conhecida como dendrograma, que mostra a relação de similaridade entre os objetos.

Existem dois tipos principais de classificação hierárquica: aglomerativa e divisiva. O método aglomerativo é mais comum e começa com cada objeto em seu próprio cluster, e sucessivamente agrupa os objetos mais próximos, enquanto o

método divisivo começa com todos os objetos em um único cluster e sucessivamente divide-os em grupos menores.

Análises de classificação são amplamente utilizadas em várias áreas, incluindo bioinformática, análise de dados espaciais, agronomia, entre outros. Além disso, pode ser utilizada com diferentes medidas de distância e diferentes métodos de agrupamento, como, por exemplo, o método “complete-linkage”, “single-linkage”, “average-linkage”, entre outros.

A Análise de Classificação Hierárquica é uma técnica exploratória importante que pode ajudar a identificar padrões e relações em dados não estruturados. No entanto, é importante lembrar que a interpretação dos resultados pode ser subjetiva e depender do contexto da aplicação.

17.1 Organização básica

```
dev.off() #apaga os graficos, se houver algum
rm(list=ls(all=TRUE)) ##LIMPA A MEMORIA
cat("\014") #limpa o console
#rm(list=ls(all=TRUE)); cat("\014"); dev.off() # tudo na mesma linha
```

17.1.1 Pacotes

Instalando os pacotes necessários para esse módulo

```
install.packages("tidyverse")
install.packages("openxlsx")
install.packages("vegan")
install.packages("dplyr")
install.packages("RColorBrewer")
install.packages("gplots")
```

```
library(tidyverse)
```

Os códigos acima, são usados para instalar os pacotes necessários para este módulo. Esses códigos são comandos para instalar pacotes no R. Um pacote é uma coleção de funções, dados e documentação que ampliam as capacidades do R (R CRAN e RStudio). Depois de instalar um pacote, você precisa carregá-lo na sua sessão R com a função `library()`. Por exemplo, no código acima, carregamos o pacote `tidyverse`, usando a função `library(openxlsx)`. Isso irá permitir que você use as funções do pacote na sua sessão R. Você precisa carregar um pacote toda vez que iniciar uma nova sessão R e quiser usar um pacote instalado. Os demais pacotes instalados serão carregados ao longo desse tutorial a medida que cada pacote for sendo necessário.

Agora vamos **definir o diretório de trabalho**. Esse código é usado para obter e definir o diretório de trabalho atual no R. O comando `getwd()` retorna o caminho do diretório onde o R está lendo e salvando arquivos. O comando `setwd()` muda esse diretório de trabalho para o caminho especificado entre aspas. No seu caso, você deve ajustar o caminho para o seu próprio diretório de trabalho. **Lembre de usar a barra “/” entre os diretórios. E não a contra-barra “\”.**

Definindo o diretório de trabalho e instalando os pacotes necessários:

```
getwd()
setwd("C:/Seu/Diretório/De/Trabalho")
```

Alternativamente você pode ir na barra de tarefas e escolhes as opções:
SESSION -> SET WORKING DIRECTORY -> CHOOSE DIRECTORY

17.1.2 Sobre os dados do PPBio

A planilha `ppbio` contém os dados de abundância de espécies em diferentes unidades amostrais (UA's) (Figura 17.1 (Veja Programa de Pesquisa em Biodiversidade – PPBio)). Essa é a **matriz bruta de dados**, porque os valores ainda não foram ajustados para os valores de Captura Por Unidade de Esforço (CPUE), nem foram relativizados ou transformados.

17.1.3 Importando a planilha de trabalho

Note que o símbolo `#` em programação R significa que o texto que vem depois dele é um comentário e não será executado pelo programa. Isso é útil para explicar o código ou deixar anotações. Ajuste a segunda linha do código abaixo para refletir “C:/Seu/Diretório/De/Trabalho/Planilha.xlsx”.

```
library(openxlsx)
ppbio <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/ppbio06p-peixes.xlsx",
                    rowNames = T,
                    colNames = T,
                    sheet = "Sheet1")
str(ppbio)
ppbio_ma <- as.matrix(ppbio) #lê ppbio como uma matriz
str(ppbio_ma)
#ppbio
#ppbio_ma
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	sitio	ap-davis	as-bimac	as-fasci	ch-bimac	ci-ocela	ci-orien	co-macro	co-heter	cr-menez	cu-lepid	cy	de
2	S-A-ZA1	0	1	0	0	0	0	0	0	0	0	0	0
3	S-R-CC1	0	99	0	0	0	0	0	0	0	0	0	0
4	S-R-CT1	0	194	55	0	0	5	0	1	14	0	0	0
5	S-R-CP1	0	19	0	0	0	0	0	0	0	0	0	0
6	S-A-TA1	0	23	1	13	0	0	0	0	0	0	0	0
7	S-R-CT2	0	142	3	3	0	69	0	0	0	4	0	0
8	S-R-CP2	0	5	1	0	40	9	0	0	0	0	0	0
9	S-A-TA2	0	46	0	178	0	0	0	0	0	0	0	0
10	S-R-CT3	0	206	64	0	0	25	0	0	0	8	0	0
11	S-R-CP3	0	16	0	0	13	24	0	0	0	0	0	0
12	S-A-TA3	0	234	7	238	0	0	2	0	0	0	0	0
13	S-R-CT4	0	0	1	0	0	5	0	0	0	1	0	0
14	S-R-CP4	0	0	0	0	11	6	0	0	0	0	0	0
15	S-A-TA4	0	394	0	273	0	0	0	0	0	1	0	0
16	B-A-MU1	0	12	0	0	0	0	0	0	0	0	0	0
17	B-R-ET1	0	3	0	0	0	0	0	0	0	0	0	0
18	B-A-GU1	0	2	2	0	0	0	0	0	0	0	0	0
19	B-R-PC2	5	44	0	0	2	0	0	0	0	0	0	0
20	B-A-MU2	0	99	0	0	0	0	0	0	0	0	0	0
21	B-A-GU2	0	0	0	0	0	0	0	0	0	0	0	0
22	B-R-PC3	22	75	7	0	4	0	0	0	0	0	0	21
23	B-A-MU3	0	511	0	0	0	0	0	0	0	0	0	0
24	B-A-GU3	0	6	0	0	0	0	0	0	0	0	0	0
25	B-R-PC4	0	7	17	0	0	0	0	0	0	0	0	0
26	B-A-MU4	0	235	0	0	0	0	0	0	0	0	0	0
27	B-A-GU4	0	13	0	0	0	0	0	0	0	0	0	0

Figure 17.1: Parte da planilha de dados brutos do PPBio.

17.2 Reset point

ATENÇÃO Aqui substitui-se uma nova matriz de dados, relativizada e/ou transformada, pela matriz de trabalho inicial.

```
m_trab <- (ppbio) # <1>
```

1. Aqui usaremos as matrizes transposta/relativizada/transformada/particionada

17.2.0.1 Outra forma de achar e importar uma planilha

```
getwd()
ppbio <- read.xlsx(file.choose(),
                    rowNames = T, colNames = T,
                    sheet = "Sheet1")
```

17.3 Nomenclatura das matrizes em AMD

No tocante aos tipos de atributos e seu tratamento pré-análises, uma matriz de dados pode ser dos tipos apresentados na tabela 17.1, abaixo.

```
## Warning in table_info$align_vector[column] <-
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para
## para substituir não é um múltiplo do comprimento do substituto
```

17.4 Cluster 1: Piloto automático

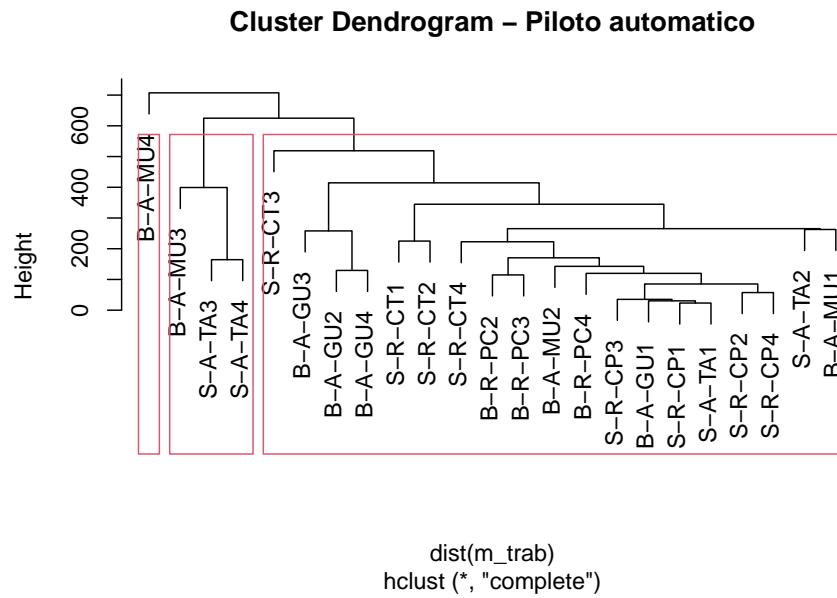
ATENÇÃO NÃO RECOMENDADO

Aqui fazemos uma Análise Cluster no piloto automático. Não decidimos nenhum dos parâmetros importantes para uma Classificação.

```
cluster1 <- hclust(dist(m_trab))
plot(cluster1, main = "Cluster Dendrogram - Piloto automatico")
rect.hclust(cluster1, k=3, h = NULL)
#?dist
#?hclust
```

Table 17.1: Nomenclatura das matrizes em AMD em relação aos atributos das colunas.

Nome	Atributos (colunas)
Matriz comunitaria	Os atributos são táxons (ex. espécies, gêneros, morfotipos)
Matriz ambiental	Os atributos são dados ambientais (ex. pH, condutividade, temperatura)
Matriz bruta	Os atributos ainda não receberam nenhum tipo de tratamento estatístico (valores brutos, como coletados)
Matriz transposta	Os atributos foram transpostos para as linhas
Matriz relativizada	Os atributos foram relativizados por um critério de tamanho ou de variação (ex. dividir os valores de cada coluna pela soma)
Matriz transformada	Foi aplicado um operador matemático a todos os atributos (ex. raiz quadrada, log)



É possível definir uma caixa mostrando os principais grupos formados. Isso pode ser feito estabelecendo quantos grupos se quer mostrar, usando a função `k = no. de grupos`. Ou pode-se definir os grupos formados até uma determinada distância, usando a função `h = altura dos grupos` no eixo das distâncias.

17.4.1 Histórico das fusões

```

library(dplyr)
cluster1$merge #mostra o histórico das fusões
idrow <- mutate(m_trab, id = row_number()) #cria um df com os numeros das linhas
idrow %>% relocate(id)
#?relocate

##      [,1] [,2]
## [1,]   -2   -3
## [2,]  -14    1
## [3,]   -8    2
## [4,]   -5  -11
## [5,]    3    4
## [6,]  -15  -18
## [7,]  -21    5
## [8,]  -17  -23
## [9,]  -16    7
## [10,]   -9  -12
  
```

390 CHAPTER 17. R MÓDULO 7 - ANÁLISE DE CLASSIFICAÇÃO HIERÁRQUICA - SAHN

```

## [11,]   6   9
## [12,] -10  11
## [13,]  -1  -4
## [14,] -20   8
## [15,]  -6 -13
## [16,]  12  15
## [17,]  13  16
## [18,] -19  10
## [19,]  14  17
## [20,]  -7  19
## [21,]  18  20
## [22,] -22  21
##           id ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro
## S-R-CT1  1      0     194      55      0      0      5      0
## S-R-CP1  2      0      19       0      0      0      0      0
## S-A-TA1  3      0     23       1     13      0      0      0
## S-R-CT2  4      0    142       3       3      0     69      0
## S-R-CP2  5      0      5       1      0     40      9      0
## S-A-TA2  6      0     46       0    178      0      0      0
## S-R-CT3  7      0    206      64      0      0     25      0
## S-R-CP3  8      0     16       0      0     13     24      0
## S-A-TA3  9      0    234       7    238      0      0      2
## S-R-CT4 10      0      0       1      0      0      5      0
## S-R-CP4 11      0      0       0      0     11      6      0
## S-A-TA4 12      0    394       0    273      0      0      0
## B-A-MU1 13      0     12       0      0      0      0      0
## B-A-GU1 14      0      2       2      0      0      0      0
## B-R-PC2 15      5     44       0      0      2      0      0
## B-A-MU2 16      0     99       0      0      0      0      0
## B-A-GU2 17      0      0       0      0      0      0      0
## B-R-PC3 18     22     75       7      0      4      0      0
## B-A-MU3 19      0    511       0      0      0      0      0
## B-A-GU3 20      0      6       0      0      0      0      0
## B-R-PC4 21      0      7     17      0      0      0      0
## B-A-MU4 22      0    235       0      0      0      0      0
## B-A-GU4 23      0     13       0      0      0      0      0
##           co-heter cr-menez cu-lepid cy-gilbe ge-brasi he-margi ho-malab hy-pusar
## S-R-CT1      1     14       0       0      3      0      1      9
## S-R-CP1      0      0       0       0      0      0      5      2
## S-A-TA1      0      0       0       0      0      0      0      0
## S-R-CT2      0      4       0       0      0      1     17     43
## S-R-CP2      0      0       0       0      0      0     10      2
## S-A-TA2      0      0       0       0      0      0      2      0
## S-R-CT3      0      8       0       0      1      0     31     11
## S-R-CP3      0      0       0       0      0      0      4      0
## S-A-TA3      0      0       0       0      0      0     20      0

```

## S-R-CT4	0	1	0	50	3	1	4	3
## S-R-CP4	0	0	0	0	0	0	2	0
## S-A-TA4	0	1	0	0	1	0	9	0
## B-A-MU1	0	0	0	0	190	0	0	0
## B-A-GU1	0	0	0	0	7	0	0	0
## B-R-PC2	0	0	0	0	8	0	0	0
## B-A-MU2	0	0	0	0	67	0	1	0
## B-A-GU2	0	0	0	0	23	0	0	0
## B-R-PC3	0	0	21	0	16	0	2	1
## B-A-MU3	0	0	0	0	145	0	0	0
## B-A-GU3	0	0	0	0	32	0	0	0
## B-R-PC4	0	0	0	81	5	0	1	0
## B-A-MU4	0	0	0	0	509	0	0	0
## B-A-GU4	0	0	0	0	10	0	0	0
## le-melan le-piau le-taeni mo-costa mo-lepid or-nilot pa-manag pimel-sp								
## S-R-CT1	0	3	0	0	39	36	0	6
## S-R-CP1	0	0	0	0	0	0	0	0
## S-A-TA1	0	0	0	0	0	0	0	0
## S-R-CT2	0	1	0	0	1	77	0	0
## S-R-CP2	0	3	0	0	0	0	0	0
## S-A-TA2	0	0	0	0	0	0	0	0
## S-R-CT3	0	2	0	0	0	138	0	0
## S-R-CP3	0	1	0	0	0	0	0	0
## S-A-TA3	0	0	0	0	0	0	0	0
## S-R-CT4	0	0	0	0	0	73	0	0
## S-R-CP4	0	2	0	0	0	0	0	0
## S-A-TA4	0	2	0	0	0	1	0	0
## B-A-MU1	0	0	0	0	0	6	0	0
## B-A-GU1	0	0	0	0	0	3	11	0
## B-R-PC2	2	0	1	0	0	5	0	0
## B-A-MU2	0	0	0	0	0	1	0	0
## B-A-GU2	0	0	0	0	0	36	102	0
## B-R-PC3	0	0	0	1	0	65	0	0
## B-A-MU3	0	0	0	0	0	11	0	0
## B-A-GU3	0	0	0	0	0	247	250	0
## B-R-PC4	0	1	0	0	0	9	0	0
## B-A-MU4	0	0	0	0	0	1	0	0
## B-A-GU4	0	0	0	0	0	129	190	0
## po-retic po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba								
## S-R-CT1	0	47	5	0	0	40	68	
## S-R-CP1	0	15	0	0	0	14	0	
## S-A-TA1	0	0	1	0	0	4	0	
## S-R-CT2	20	221	15	0	0	60	0	
## S-R-CP2	0	32	5	0	0	0	0	
## S-A-TA2	0	0	2	0	0	0	0	
## S-R-CT3	5	326	164	1	1	38	0	

392 CHAPTER 17. R MÓDULO 7 - ANÁLISE DE CLASSIFICAÇÃO HIERÁRQUICA - SAHN

	0	10	0	0	0	0	0
## S-R-CP3	0	10	0	0	0	0	0
## S-A-TA3	0	0	0	0	0	0	0
## S-R-CT4	0	28	59	0	0	3	0
## S-R-CP4	0	80	0	0	0	3	0
## S-A-TA4	0	0	3	0	0	0	0
## B-A-MU1	0	0	0	0	0	0	0
## B-A-GU1	0	0	0	0	0	0	0
## B-R-PC2	0	0	9	0	0	10	0
## B-A-MU2	10	8	0	0	0	0	0
## B-A-GU2	0	0	0	0	0	0	0
## B-R-PC3	0	0	6	0	0	93	0
## B-A-MU3	46	48	1	0	0	0	0
## B-A-GU3	0	0	0	0	0	0	0
## B-R-PC4	0	0	0	0	0	31	0
## B-A-MU4	266	163	0	0	0	0	0
## B-A-GU4	0	0	0	0	0	0	0
	se-spilo st-noton sy-marmo te-chalc tr-signa						
## S-R-CT1	0	1	0	0	18		
## S-R-CP1	0	0	0	0	0		
## S-A-TA1	0	0	0	0	0		
## S-R-CT2	0	25	0	0	15		
## S-R-CP2	0	0	1	0	0		
## S-A-TA2	0	0	0	0	0		
## S-R-CT3	1	115	0	0	7		
## S-R-CP3	0	0	0	0	0		
## S-A-TA3	0	0	0	0	0		
## S-R-CT4	0	64	0	0	141		
## S-R-CP4	0	0	0	0	0		
## S-A-TA4	0	0	0	0	0		
## B-A-MU1	0	0	0	0	0		
## B-A-GU1	0	0	0	0	0		
## B-R-PC2	0	0	0	76	23		
## B-A-MU2	0	0	0	0	0		
## B-A-GU2	0	0	0	0	0		
## B-R-PC3	0	0	0	58	0		
## B-A-MU3	0	0	0	0	0		
## B-A-GU3	0	0	0	0	0		
## B-R-PC4	0	0	0	0	4		
## B-A-MU4	0	0	0	0	0		
## B-A-GU4	0	0	0	0	0		

17.4.1.1 Alguns ajustes para melhorar a visualização

```

merge <- as.data.frame(cluster1$merge)
merge[nrow(merge)+1,] = c("0","0")
merge
uas <- as.data.frame(rownames_to_column(m_trab, var = "UAs"))
uas
merges <- cbind(uas[c("UAs")], merge)
merges

```

```

##      V1   V2
## 1    -2   -3
## 2   -14    1
## 3   -8    2
## 4   -5  -11
## 5    3    4
## 6  -15  -18
## 7  -21    5
## 8  -17  -23
## 9  -16    7
## 10   -9  -12
## 11   6    9
## 12  -10   11
## 13   -1   -4
## 14  -20    8
## 15   -6  -13
## 16   12   15
## 17   13   16
## 18  -19   10
## 19   14   17
## 20   -7   19
## 21   18   20
## 22  -22   21
## 23    0    0
##           UAs ap-davis as-bimac as-fasci ch-bimac ci-ocela ci-orien co-macro
## 1 S-R-CT1       0     194      55       0       0       5       0
## 2 S-R-CP1       0      19       0       0       0       0       0
## 3 S-A-TA1       0     23       1      13       0       0       0
## 4 S-R-CT2       0    142       3       3       0      69       0
## 5 S-R-CP2       0      5       1       0      40       9       0
## 6 S-A-TA2       0     46       0     178       0       0       0
## 7 S-R-CT3       0    206      64       0       0      25       0
## 8 S-R-CP3       0     16       0       0      13      24       0
## 9 S-A-TA3       0    234       7     238       0       0       2
## 10 S-R-CT4      0      0       1       0       0       5       0
## 11 S-R-CP4      0      0       0       0      11       6       0

```

394 CHAPTER 17. R MÓDULO 7 - ANÁLISE DE CLASSIFICAÇÃO HIERÁRQUICA - SAHN

## 12 S-A-TA4	0	394	0	273	0	0	0
## 13 B-A-MU1	0	12	0	0	0	0	0
## 14 B-A-GU1	0	2	2	0	0	0	0
## 15 B-R-PC2	5	44	0	0	2	0	0
## 16 B-A-MU2	0	99	0	0	0	0	0
## 17 B-A-GU2	0	0	0	0	0	0	0
## 18 B-R-PC3	22	75	7	0	4	0	0
## 19 B-A-MU3	0	511	0	0	0	0	0
## 20 B-A-GU3	0	6	0	0	0	0	0
## 21 B-R-PC4	0	7	17	0	0	0	0
## 22 B-A-MU4	0	235	0	0	0	0	0
## 23 B-A-GU4	0	13	0	0	0	0	0
## co-heter	cr-menez	cu-lepid	cy-gilbe	ge-brasi	he-margi	ho-malab	hy-pusar
## 1	1	14	0	0	3	0	1
## 2	0	0	0	0	0	0	5
## 3	0	0	0	0	0	0	0
## 4	0	4	0	0	0	1	17
## 5	0	0	0	0	0	0	10
## 6	0	0	0	0	0	0	2
## 7	0	8	0	0	1	0	31
## 8	0	0	0	0	0	0	4
## 9	0	0	0	0	0	0	20
## 10	0	1	0	50	3	1	4
## 11	0	0	0	0	0	0	2
## 12	0	1	0	0	1	0	9
## 13	0	0	0	0	190	0	0
## 14	0	0	0	0	7	0	0
## 15	0	0	0	0	8	0	0
## 16	0	0	0	0	67	0	1
## 17	0	0	0	0	23	0	0
## 18	0	0	21	0	16	0	2
## 19	0	0	0	0	145	0	0
## 20	0	0	0	0	32	0	0
## 21	0	0	0	81	5	0	1
## 22	0	0	0	0	509	0	0
## 23	0	0	0	0	10	0	0
## le-melan	le-piau	le-taeni	mo-costa	mo-lepid	or-nilot	pa-manag	pimel-sp
## 1	0	3	0	0	39	36	0
## 2	0	0	0	0	0	0	0
## 3	0	0	0	0	0	0	0
## 4	0	1	0	0	1	77	0
## 5	0	3	0	0	0	0	0
## 6	0	0	0	0	0	0	0
## 7	0	2	0	0	0	138	0
## 8	0	1	0	0	0	0	0
## 9	0	0	0	0	0	0	0

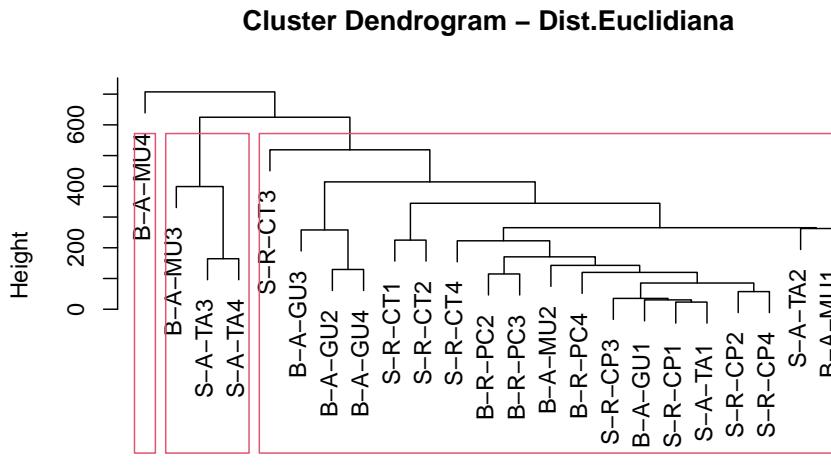
## 10	0	0	0	0	0	73	0	0
## 11	0	2	0	0	0	0	0	0
## 12	0	2	0	0	0	1	0	0
## 13	0	0	0	0	0	6	0	0
## 14	0	0	0	0	0	3	11	0
## 15	2	0	1	0	0	5	0	0
## 16	0	0	0	0	0	1	0	0
## 17	0	0	0	0	0	36	102	0
## 18	0	0	0	1	0	65	0	0
## 19	0	0	0	0	0	11	0	0
## 20	0	0	0	0	0	247	250	0
## 21	0	1	0	0	0	9	0	0
## 22	0	0	0	0	0	1	0	0
## 23	0	0	0	0	0	129	190	0
## po-retic po-vivip pr-brevi ps-rhomb ps-genise se-heter se-piaba se-spilo								
## 1	0	47	5	0	0	40	68	0
## 2	0	15	0	0	0	14	0	0
## 3	0	0	1	0	0	4	0	0
## 4	20	221	15	0	0	60	0	0
## 5	0	32	5	0	0	0	0	0
## 6	0	0	2	0	0	0	0	0
## 7	5	326	164	1	1	38	0	1
## 8	0	10	0	0	0	0	0	0
## 9	0	0	0	0	0	0	0	0
## 10	0	28	59	0	0	3	0	0
## 11	0	80	0	0	0	3	0	0
## 12	0	0	3	0	0	0	0	0
## 13	0	0	0	0	0	0	0	0
## 14	0	0	0	0	0	0	0	0
## 15	0	0	9	0	0	10	0	0
## 16	10	8	0	0	0	0	0	0
## 17	0	0	0	0	0	0	0	0
## 18	0	0	6	0	0	93	0	0
## 19	46	48	1	0	0	0	0	0
## 20	0	0	0	0	0	0	0	0
## 21	0	0	0	0	0	31	0	0
## 22	266	163	0	0	0	0	0	0
## 23	0	0	0	0	0	0	0	0
## st-noton sy-marmo te-chalc tr-signa								
## 1	1	0	0	18				
## 2	0	0	0	0				
## 3	0	0	0	0				
## 4	25	0	0	15				
## 5	0	1	0	0				
## 6	0	0	0	0				
## 7	115	0	0	7				

396 CHAPTER 17. R MÓDULO 7 - ANÁLISE DE CLASSIFICAÇÃO HIERÁRQUICA - SAHN

```
## 8      0      0      0      0
## 9      0      0      0      0
## 10     64     0      0    141
## 11     0      0      0      0
## 12     0      0      0      0
## 13     0      0      0      0
## 14     0      0      0      0
## 15     0      0    76    23
## 16     0      0      0      0
## 17     0      0      0      0
## 18     0      0    58      0
## 19     0      0      0      0
## 20     0      0      0      0
## 21     0      0      0      4
## 22     0      0      0      0
## 23     0      0      0      0
##          UAs  V1  V2
## 1  S-R-CT1 -2  -3
## 2  S-R-CP1 -14   1
## 3  S-A-TA1 -8   2
## 4  S-R-CT2 -5 -11
## 5  S-R-CP2  3   4
## 6  S-A-TA2 -15 -18
## 7  S-R-CT3 -21   5
## 8  S-R-CP3 -17 -23
## 9  S-A-TA3 -16   7
## 10 S-R-CT4 -9 -12
## 11 S-R-CP4  6   9
## 12 S-A-TA4 -10  11
## 13 B-A-MU1 -1  -4
## 14 B-A-GU1 -20   8
## 15 B-R-PC2 -6 -13
## 16 B-A-MU2 12  15
## 17 B-A-GU2 13  16
## 18 B-R-PC3 -19  10
## 19 B-A-MU3 14  17
## 20 B-A-GU3 -7  19
## 21 B-R-PC4 18  20
## 22 B-A-MU4 -22  21
## 23 B-A-GU4   0   0
```

17.5 Cluster 2: Criando a matriz de distâncias Euclidiana

```
m_dists <- dist(m_trab, method = "euclidean",
                  diag = TRUE,
                  upper = FALSE)
#m_dists
cluster2 <- hclust(m_dists, method = "complete") #método de fusão
plot (cluster2, main = "Cluster Dendrogram - Dist.Euclidiana")
rect.hclust(cluster2, k = 3)
#?dist
#?hclust
```



m_dists
hclust (*, "complete")

O significado do argumento `method` = da função `dist()` usada no código acima, depende do contexto em que ela é usada.

- A função `dist()` no R é usada para calcular as distâncias entre as observações em um conjunto de dados. Essa função tem vários métodos de cálculo de distância, que podem ser especificados pelo argumento `method` =, a exemplo de “euclidean”, “manhattan”, “canberra”, entre outras.
- Quando o argumento `method` = é usado em conjunto com a função `hclust()`, como no código acima, para realizar uma análise de classificação (ou cluster) hierárquica, então ela passa a representar um método de fusão, a exemplo de UPGMA, “nearest neighbour”, “furthest

Table 17.2: Métodos de fusão disponíveis com o argumento ‘method=’ da função ‘`hclust`’: Lembrando que UPGMA significa ”Unweighted Pairwise Group Method with Arithmetic Mean”

Method=	Método de fusão
ward.D	Método de Ward com variância mínima
ward.D2	Método de Ward com variância mínima ajustada
single	Método de ligação simples (vizinho mais próximo)
complete	Método de ligação completa (vizinho mais distante)
average	Método de ligação baseado na média, equivale ao UPGMA
mcquitty	Método de ligação de McQuitty, equivale ao WPGMA
median	Método de ligação baseado na mediana, ou WPGMC
centroid	Método de ligação baseado no centróide, ou UPGMC

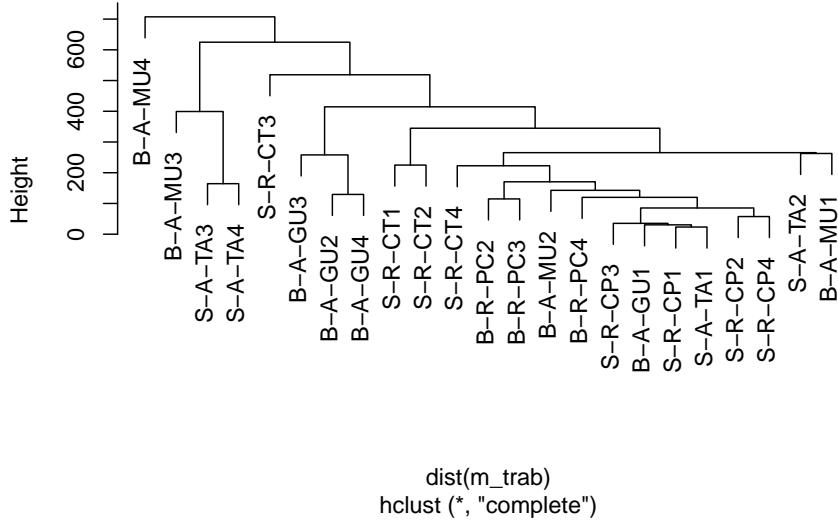
neighbour”, entre outros.

A tabela 17.2 resume os métodos de fusão disponíveis na função `hclust()`.

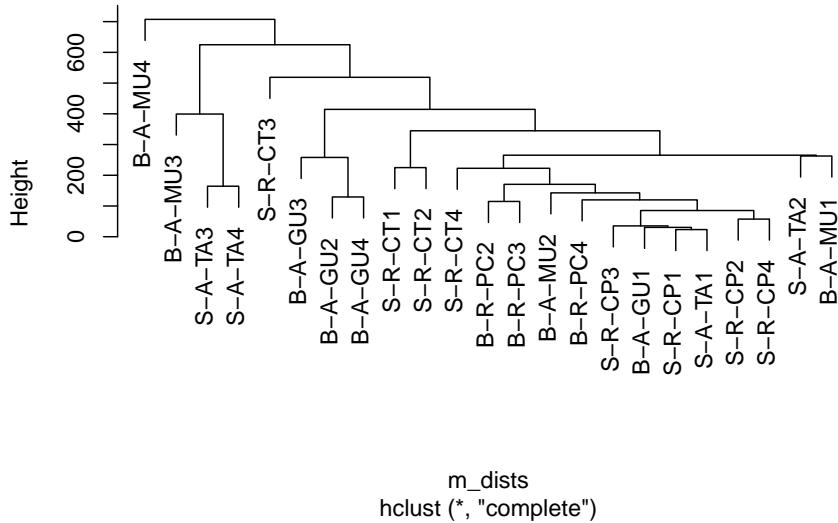
```
## Warning in table_info$align_vector[column] <-
## unlist(lapply(table_info$align_vector_origin[column], : número de itens para
## para substituir não é um múltiplo do comprimento do substituto
```

```
#dev.off() #limpa os gráficos
par(mfrow = c(2,1))
plot(cluster1, main = "Cluster Dendrogram - Piloto automático")
plot(cluster2, main = "Cluster Dendrogram - Dist. Euclidiana")
```

Cluster Dendrogram – Piloto automático



Cluster Dendrogram – Dist. Euclidiana



```
par(mfrow = c(1,1))
#dev.off()
```

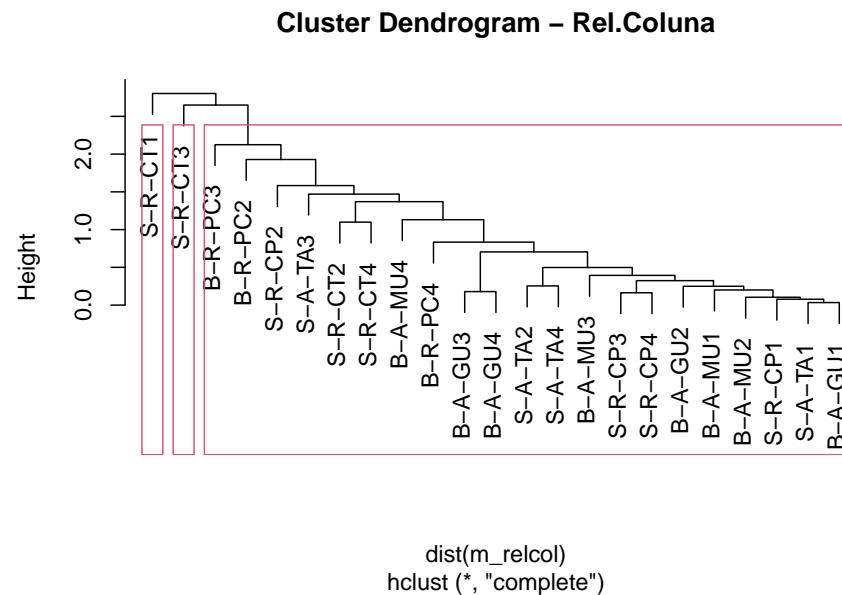
17.6 Cluster 3: Relativização pelo total da coluna

```
library(vegan)
m_relcol <- decostand(m_trab,
                       method="total", ### OUTROS METODOS: total, max, normalize, range
                       MARGIN = 2) #1-linha, 2-coluna
#View(m_relcol)
#m_relcol
```

17.6.1 Gráfico de clustering relativizado

Agora fazemos a classificação da matriz bruta após ter sido relativizada pelo total das colunas.

```
cluster3 <- hclust(dist(m_relcol))
plot(cluster3, main = "Cluster Dendrogram - Rel.Coluna")
rect.hclust(cluster3, k = 3)
```



17.7. CLUSTER 4: TRANSFORMAÇÃO PELO ARCOSENO DA RAIZ QUADRADA401

17.7 Cluster 4: Transformação pelo arcoseno da raiz quadrada

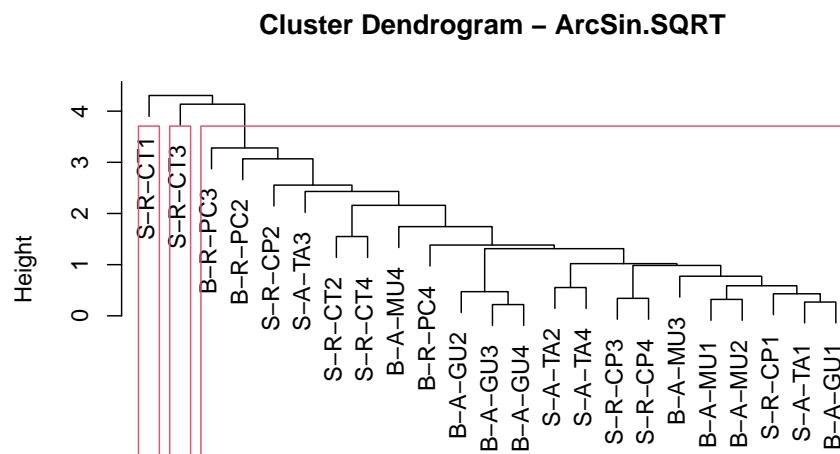
Agora aplicaremos a transformação do arcoseno da raiz quadrada na matriz relativizada por colunas. Lembre que em ambas as matrizes entramos dados que variam entre 0 e 1, e os resultados retornados também variam entre 0 e 1, mas em um nível de compressão de valores extremos mais desejável. Usaremos a matriz relativizada/transformada para fazermos a proxima classificação.

```
m_trns <- asin(sqrt(m_relc)) #valores de entrada tem que ser entre 0-1
#View(m_trns)
#m_trns
```

17.7.1 Gráfico de clustering relativizado/transformado

Aqui fazemos uma Análise Cluster no piloto automático, para comparar com os resultados da primeira classificação, que não havia sido relativizada/transformada.

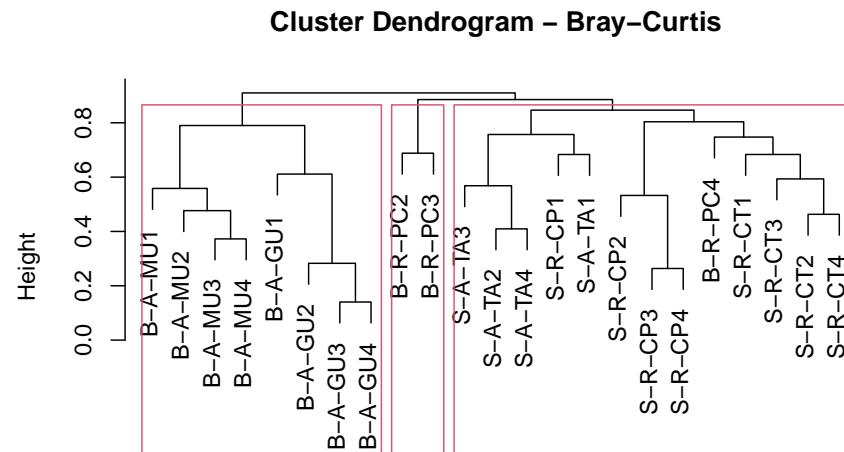
```
cluster4 <- hclust(dist(m_trns))
plot(cluster4, main = "Cluster Dendrogram - ArcSin.SQRT")
rect.hclust(cluster4, k = 3)
```



```
dist(m_trns)
hclust (*, "complete")
```

17.8 Cluster 5: Controle total dos parâmetros

```
vegdist <- vegdist(m_trns, method = "bray",
                    diag = TRUE,
                    upper = FALSE)
as.matrix(vegdist)[1:7, 1:7]
cluster5 <- hclust(vegdist, method = "average")
plot(cluster5, main = "Cluster Dendrogram - Bray-Curtis")
rect.hclust(cluster5, k = 3, h = NULL)
```



```
vegdist
hclust (*, "average")
```

```
#h = 0.8 fornece os grupos formados na altura h
#vegdist
cluster5$merge
cluster5$height
idrow <- mutate(m_trns, id = row_number())
idrow %>% relocate(id)
```

```
##          S-R-CT1   S-R-CP1   S-A-TA1   S-R-CT2   S-R-CP2   S-A-TA2   S-R-CT3
## S-R-CT1 0.0000000 0.8743721 0.9338269 0.6274997 0.8106894 0.9420728 0.6582219
## S-R-CP1 0.8743721 0.0000000 0.6833816 0.7759468 0.7726098 0.7342613 0.8672111
## S-A-TA1 0.9338269 0.6833816 0.0000000 0.8789631 0.9178304 0.5700984 0.9404687
## S-R-CT2 0.6274997 0.7759468 0.8789631 0.0000000 0.7280378 0.8836068 0.5169933
## S-R-CP2 0.8106894 0.7726098 0.9178304 0.7280378 0.0000000 0.8915271 0.8012846
## S-A-TA2 0.9420728 0.7342613 0.5700984 0.8836068 0.8915271 0.0000000 0.9412636
```

```

## S-R-CT3 0.6582219 0.8672111 0.9404687 0.5169933 0.8012846 0.9412636 0.0000000
##      [,1] [,2]
## [1,] -20  -23
## [2,]  -8  -11
## [3,] -17   1
## [4,] -19  -22
## [5,]  -6  -12
## [6,]  -4  -10
## [7,] -16    4
## [8,]  -5    2
## [9,] -13    7
## [10,]  -9    5
## [11,]  -7    6
## [12,] -14    3
## [13,]  -2   -3
## [14,]  -1   11
## [15,] -15  -18
## [16,] -21   14
## [17,]  10   13
## [18,]   9   12
## [19,]   8   16
## [20,]  17   19
## [21,]  15   20
## [22,]  18   21
## [1] 0.1402400 0.2637008 0.2827470 0.3726150 0.4095558 0.4631145 0.4766705
## [8] 0.5326053 0.5584043 0.5683644 0.5934519 0.6112108 0.6833816 0.6834576
## [15] 0.6881517 0.7475412 0.7570089 0.7899914 0.8037471 0.8465927 0.8855551
## [22] 0.9099458
##      id ap-davis  as-bimac  as-fasci  ch-bimac  ci-ocela  ci-orien
## S-R-CT1 1 0.000000 0.29580113 0.63106017 0.00000000 0.0000000 0.1880966
## S-R-CP1 2 0.000000 0.09135411 0.00000000 0.00000000 0.0000000 0.0000000
## S-A-TA1 3 0.000000 0.10054093 0.07963989 0.13621375 0.0000000 0.0000000
## S-R-CT2 4 0.000000 0.25205752 0.13823439 0.06527916 0.0000000 0.7679121
## S-R-CP2 5 0.000000 0.04681561 0.07963989 0.00000000 0.8570719 0.2535816
## S-A-TA2 6 0.000000 0.14242797 0.00000000 0.52646034 0.0000000 0.0000000
## S-R-CT3 7 0.000000 0.30509802 0.68988155 0.00000000 0.0000000 0.4313758
## S-R-CP3 8 0.000000 0.08381382 0.00000000 0.00000000 0.4455406 0.4220958
## S-A-TA3 9 0.000000 0.32588896 0.21207071 0.61998604 0.0000000 0.0000000
## S-R-CT4 10 0.000000 0.00000000 0.07963989 0.00000000 0.0000000 0.1880966
## S-R-CP4 11 0.000000 0.00000000 0.00000000 0.00000000 0.4076059 0.2062968
## S-A-TA4 12 0.000000 0.42841263 0.00000000 0.67165367 0.0000000 0.0000000
## B-A-MU1 13 0.000000 0.07256361 0.00000000 0.00000000 0.0000000 0.0000000
## B-A-GU1 14 0.000000 0.02960230 0.11274751 0.00000000 0.0000000 0.0000000
## B-R-PC2 15 0.444860 0.13927667 0.00000000 0.00000000 0.1698463 0.0000000
## B-A-MU2 16 0.000000 0.20977540 0.00000000 0.00000000 0.0000000 0.0000000
## B-A-GU2 17 0.000000 0.00000000 0.00000000 0.0000000 0.0000000 0.0000000

```

```

## B-R-PC3 18 1.125936 0.18225721 0.21207071 0.00000000 0.2413830 0.0000000
## B-A-MU3 19 0.000000 0.49281178 0.00000000 0.00000000 0.0000000 0.0000000
## B-A-GU3 20 0.000000 0.05128768 0.00000000 0.00000000 0.0000000 0.0000000
## B-R-PC4 21 0.000000 0.05540108 0.33420332 0.00000000 0.0000000 0.0000000
## B-A-MU4 22 0.000000 0.32661037 0.00000000 0.00000000 0.0000000 0.0000000
## B-A-GU4 23 0.000000 0.07553213 0.00000000 0.00000000 0.0000000 0.0000000
##          co-macro co-heter cr-menez cu-lepid cy-gilbe ge-brasi he-margi
## S-R-CT1 0.000000 1.570796 0.7853982 0.000000 0.0000000 0.05425923 0.0000000
## S-R-CP1 0.000000 0.000000 0.0000000 0.000000 0.00000000 0.0000000 0.0000000
## S-A-TA1 0.000000 0.000000 0.0000000 0.000000 0.00000000 0.00000000 0.0000000
## S-R-CT2 0.000000 0.000000 0.3875967 0.000000 0.0000000 0.00000000 0.7853982
## S-R-CP2 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.00000000 0.0000000
## S-A-TA2 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.00000000 0.0000000
## S-R-CT3 0.000000 0.000000 0.5639426 0.000000 0.0000000 0.03131633 0.0000000
## S-R-CP3 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.00000000 0.0000000
## S-A-TA3 1.570796 0.000000 0.0000000 0.000000 0.0000000 0.00000000 0.0000000
## S-R-CT4 0.000000 0.000000 0.1901256 0.000000 0.6659445 0.05425923 0.7853982
## S-R-CP4 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.00000000 0.0000000
## S-A-TA4 0.000000 0.000000 0.1901256 0.000000 0.0000000 0.03131633 0.0000000
## B-A-MU1 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.44626053 0.0000000
## B-A-GU1 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.08293673 0.0000000
## B-R-PC2 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.08867767 0.0000000
## B-A-MU2 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.25918549 0.0000000
## B-A-GU2 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.15073345 0.0000000
## B-R-PC3 0.000000 0.000000 0.0000000 1.570796 0.0000000 0.12557463 0.0000000
## B-A-MU3 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.38659505 0.0000000
## B-A-GU3 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.17806243 0.0000000
## B-R-PC4 0.000000 0.000000 0.0000000 0.000000 0.9048519 0.07007133 0.0000000
## B-A-MU4 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.78441777 0.0000000
## B-A-GU4 0.000000 0.000000 0.0000000 0.000000 0.0000000 0.09917726 0.0000000
##          ho-malab hy-pusar le-melan le-piau le-taeni mo-costa mo-lepid
## S-R-CT1 0.09592969 0.3640209 0.000000 0.4636476 0.000000 0.000000 1.4120161
## S-R-CP1 0.21584866 0.1686344 0.000000 0.0000000 0.000000 0.000000 0.0000000
## S-A-TA1 0.00000000 0.0000000 0.000000 0.0000000 0.000000 0.000000 0.0000000
## S-R-CT2 0.40598283 0.8918340 0.000000 0.2611574 0.000000 0.000000 0.1587802
## S-R-CP2 0.30772498 0.1686344 0.000000 0.4636476 0.000000 0.000000 0.0000000
## S-A-TA2 0.13587479 0.0000000 0.000000 0.0000000 0.000000 0.000000 0.0000000
## S-R-CT3 0.56249105 0.4045563 0.000000 0.3737922 0.000000 0.000000 0.0000000
## S-R-CP3 0.19275669 0.0000000 0.000000 0.2611574 0.000000 0.000000 0.0000000
## S-A-TA3 0.44266923 0.0000000 0.000000 0.0000000 0.000000 0.000000 0.0000000
## S-R-CT4 0.19275669 0.2070324 0.000000 0.0000000 0.000000 0.000000 0.0000000
## S-R-CP4 0.13587479 0.0000000 0.000000 0.3737922 0.000000 0.000000 0.0000000
## S-A-TA4 0.29145679 0.0000000 0.000000 0.3737922 0.000000 0.000000 0.0000000
## B-A-MU1 0.00000000 0.0000000 0.000000 0.0000000 0.000000 0.000000 0.0000000
## B-A-GU1 0.00000000 0.0000000 0.000000 0.0000000 0.000000 0.000000 0.0000000
## B-R-PC2 0.00000000 0.0000000 1.570796 0.0000000 1.570796 0.000000 0.0000000

```

```

## B-A-MU2 0.09592969 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## B-A-GU2 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## B-R-PC3 0.13587479 0.1189585 0.0000000 0.0000000 0.0000000 1.570796 0.0000000
## B-A-MU3 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## B-A-GU3 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## B-R-PC4 0.09592969 0.0000000 0.0000000 0.2611574 0.0000000 0.0000000 0.0000000
## B-A-MU4 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## B-A-GU4 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##          or-nilot pa-manag pimel-sp po-retic po-vivip pr-brevi ps-rhomb
## S-R-CT1 0.20878001 0.0000000 1.570796 0.0000000 0.22101460 0.13650631 0.0000000
## S-R-CP1 0.00000000 0.0000000 0.0000000 0.0000000 0.12416313 0.000000000 0.0000000
## S-A-TA1 0.00000000 0.0000000 0.0000000 0.0000000 0.000000000 0.06089569 0.0000000
## S-R-CT2 0.30797141 0.0000000 0.0000000 0.2424450 0.49537824 0.23794112 0.0000000
## S-R-CP2 0.00000000 0.0000000 0.0000000 0.0000000 0.18188749 0.13650631 0.0000000
## S-A-TA2 0.00000000 0.0000000 0.0000000 0.0000000 0.000000000 0.08617291 0.0000000
## S-R-CT3 0.41785958 0.0000000 0.0000000 0.1203286 0.61547971 0.89364927 1.570796
## S-R-CP3 0.00000000 0.0000000 0.0000000 0.0000000 0.10129161 0.000000000 0.0000000
## S-A-TA3 0.00000000 0.0000000 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
## S-R-CT4 0.29961011 0.0000000 0.0000000 0.0000000 0.17002156 0.48641493 0.0000000
## S-R-CP4 0.00000000 0.0000000 0.0000000 0.0000000 0.29005641 0.000000000 0.0000000
## S-A-TA4 0.03455130 0.0000000 0.0000000 0.0000000 0.000000000 0.10560544 0.0000000
## B-A-MU1 0.08471752 0.0000000 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
## B-A-GU1 0.05986846 0.1415090 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
## B-R-PC2 0.07732071 0.0000000 0.0000000 0.0000000 0.000000000 0.18360401 0.0000000
## B-A-MU2 0.03455130 0.0000000 0.0000000 0.1705861 0.09056689 0.000000000 0.0000000
## B-A-GU2 0.20878001 0.4439109 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
## B-R-PC3 0.28223829 0.0000000 0.0000000 0.0000000 0.000000000 0.14962891 0.0000000
## B-A-MU3 0.11482305 0.0000000 0.0000000 0.3726604 0.22339293 0.06089569 0.0000000
## B-A-GU3 0.57389635 0.7374041 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
## B-R-PC4 0.10381968 0.0000000 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
## B-A-MU4 0.03455130 0.0000000 0.0000000 1.0665527 0.42053434 0.000000000 0.0000000
## B-A-GU4 0.40318427 0.6263078 0.0000000 0.0000000 0.000000000 0.000000000 0.0000000
##          ps-genise se-heter se-piaba se-spilo st-noton sy-marmo te-chalc
## S-R-CT1 0.000000 0.3764349 1.570796 0.000000 0.06989994 0.000000 0.0000000
## S-R-CP1 0.000000 0.2192313 0.000000 0.000000 0.000000000 0.000000 0.0000000
## S-A-TA1 0.000000 0.1165111 0.000000 0.000000 0.000000000 0.000000 0.0000000
## S-R-CT2 0.000000 0.4670175 0.000000 0.000000 0.35673339 0.000000 0.0000000
## S-R-CP2 0.000000 0.0000000 0.000000 0.000000 0.000000000 1.570796 0.0000000
## S-A-TA2 0.000000 0.0000000 0.000000 0.000000 0.000000000 0.000000 0.0000000
## S-R-CT3 1.570796 0.3664456 0.000000 1.570796 0.84652593 0.000000 0.0000000
## S-R-CP3 0.000000 0.0000000 0.000000 0.000000 0.000000000 0.000000 0.0000000
## S-A-TA3 0.000000 0.0000000 0.000000 0.000000 0.000000000 0.000000 0.0000000
## S-R-CT4 0.000000 0.1008442 0.000000 0.000000 0.59287085 0.000000 0.0000000
## S-R-CP4 0.000000 0.1008442 0.000000 0.000000 0.000000000 0.000000 0.0000000
## S-A-TA4 0.000000 0.0000000 0.000000 0.000000 0.000000000 0.000000 0.0000000
## B-A-MU1 0.000000 0.0000000 0.000000 0.000000 0.000000000 0.000000 0.0000000

```

406 CHAPTER 17. R MÓDULO 7 - ANÁLISE DE CLASSIFICAÇÃO HIERÁRQUICA - SAHN

```

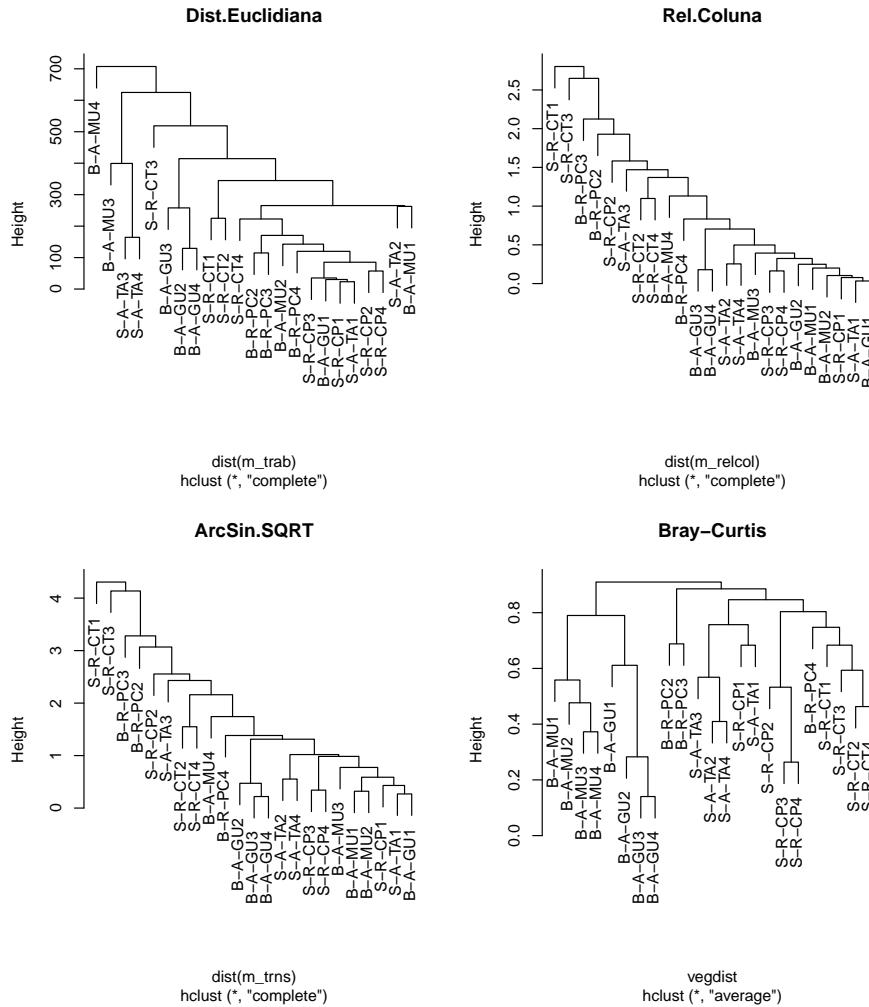
## B-A-GU1  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-R-PC2  0.000000 0.1848546 0.000000 0.000000 0.00000000 0.000000 0.8527660
## B-A-MU2  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-A-GU2  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-R-PC3  0.000000 0.5950206 0.000000 0.000000 0.00000000 0.000000 0.7180303
## B-A-MU3  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-A-GU3  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-R-PC4  0.000000 0.3295526 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-A-MU4  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
## B-A-GU4  0.000000 0.0000000 0.000000 0.000000 0.00000000 0.000000 0.0000000
##          tr-signa
## S-R-CT1  0.2985914
## S-R-CP1  0.0000000
## S-A-TA1  0.0000000
## S-R-CT2  0.2718802
## S-R-CP2  0.0000000
## S-A-TA2  0.0000000
## S-R-CT3  0.1844947
## S-R-CP3  0.0000000
## S-A-TA3  0.0000000
## S-R-CT4  0.9672667
## S-R-CP4  0.0000000
## S-A-TA4  0.0000000
## B-A-MU1  0.0000000
## B-A-GU1  0.0000000
## B-R-PC2  0.3389861
## B-A-MU2  0.0000000
## B-A-GU2  0.0000000
## B-R-PC3  0.0000000
## B-A-MU3  0.0000000
## B-A-GU3  0.0000000
## B-R-PC4  0.1391234
## B-A-MU4  0.0000000
## B-A-GU4  0.0000000

```

No código acima, $h = 0.8$ fornece os grupos formados na altura h do eixos das distâncias do dendrograma. Ou seja, no dendrograma, o eixo y (HEIGHT, "h") representa o valor da distância escolhida entre os objetos ou grupos de objetos. Portanto, se dois objetos ou grupos de objetos foram agrupados num dado valor (0.8, por exemplo) no eixo height, isso significa que a distância entre esses objetos é 0.8.

17.9 Construindo uma prancha de gráficos comparativos

```
#dev.off()
par(mfrow = c (2,2))
plot(cluster1, main = "Dist.Euclidiana")
plot(cluster3, main = "Rel.Coluna")
plot(cluster4, main = "ArcSin.SQRT")
plot(cluster5, main = "Bray-Curtis")
```



```
par(mfrow=c(1,1))
#dev.off()
```

17.10 Construindo heatmaps

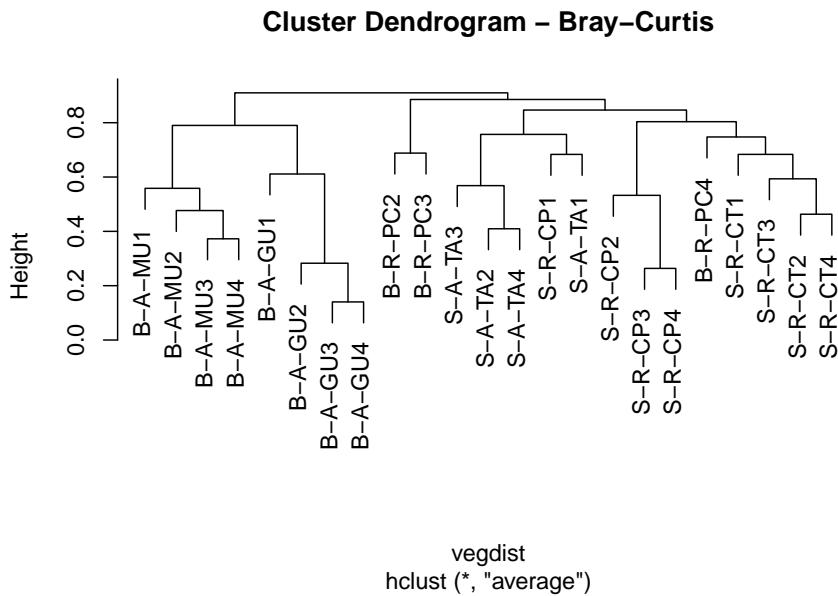
Agora que vimos de forma comparativa a importância de se relativizar e transformar nossos dados, e também de escolher de forma consciente a medida de distância e o método de fusão, podemos avançar um pouco mais na interpretação dos nossos dados.

Um “heatmap”, ou mapa de calor, é uma imagem em cores (basicamente `image(t(x))`) com um dendrograma adicionado ao lado esquerdo e/ou ao topo. Normalmente, é realizado o reordenamento das linhas e colunas de acordo com algum conjunto de valores (médias de linha ou coluna) dentro das restrições impostas pelo dendrograma.

Primeiro vamos rever o dendrograma final, usando a Distância de Bray-Curtis e o Método de Fusão UPGMA (ou “average”).

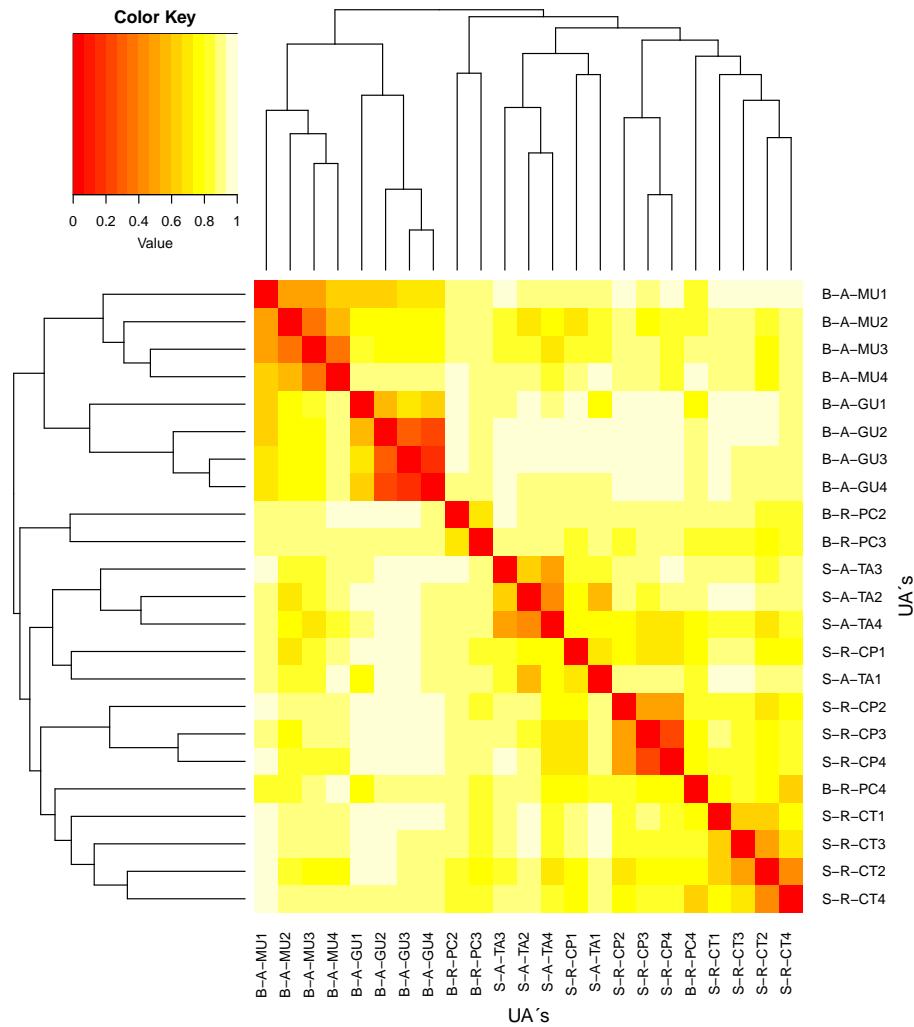
```
library("RColorBrewer")
library("gplots")
vegdist <- vegdist(m_trns, method = "bray",
                    diag = TRUE,
                    upper = FALSE)
cluster5 <- hclust(vegdist, method = "average")
plot(cluster5, main = "Cluster Dendrogram - Bray-Curtis")

heatdist <- as.matrix(vegdist)
#heatdist
```



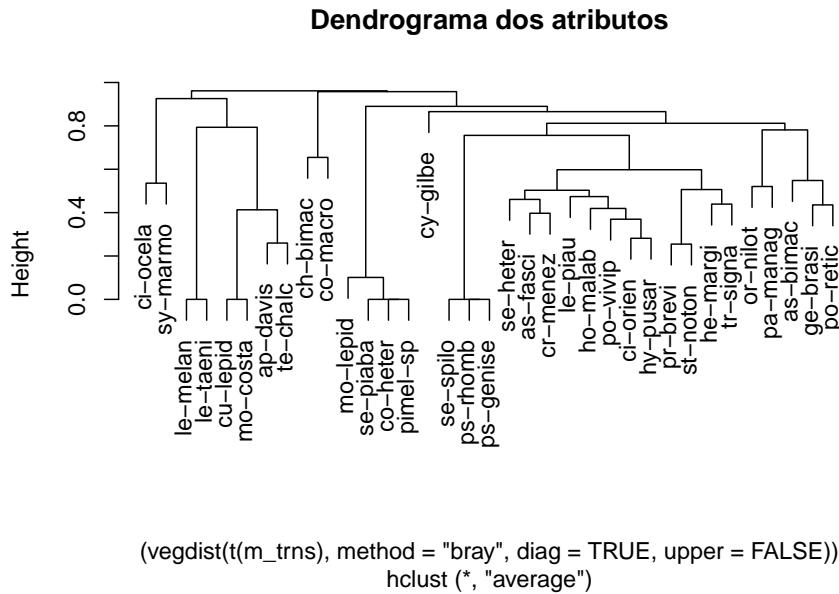
Na sequência fazemos um **heatmap objetos x objetos**, ou seja a comparação bidimensional das clusteredas entre unidades amostrais.

```
col <- rev(heat.colors(999)) #rev() reverte as cores do heatmap
heatmap.2(x=(as.matrix(vegdist)), #objetos x objetos
           Rowv = as.dendrogram(cluster5),
           Colv = as.dendrogram(cluster5),
           key = T, tracecol = NA, revC = T,
           col = heat.colors, #dissimilaridade = 1 - similaridade
           density.info = "none",
           xlab = "UA`s", ylab = "UA`s",
           mar = c(6, 6) + 0.2)
```



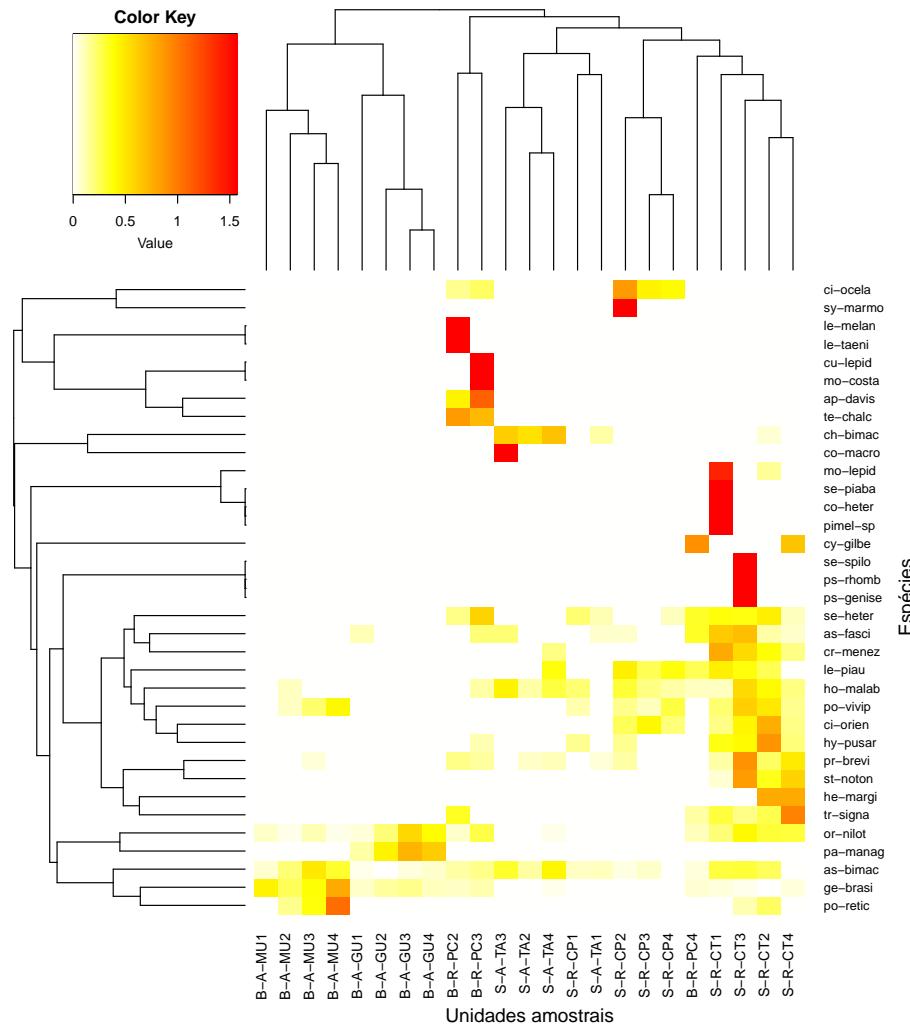
Agora vamos criar um novo dendrograma, para as espécies, mostrando sua similaridade em termos de em quais UA's elas são mais comuns.

```
cluster6 <- hclust((vegdist(t(m_trns), method = "bray",
                             diag = TRUE,
                             upper = FALSE)), method = "average")
plot(cluster6, main = "Dendrograma dos atributos")
```



Na sequencia, fazemos um novo heatmap **objetos x atributos**, com a comparação bidimensional das clustereds formadas entre unidades amostrais em relação as espécies mais comuns que ocorreram em cada unidade amostral.

```
heatmap.2(t(as.matrix(m_trns)), #objetos x atributos
          Colv = as.dendrogram(cluster5),
          Rowv = as.dendrogram(cluster6),
          key = T, tracecol = NA, revC = T,
          col = col,
          density.info = "none",
          xlab = "Unidades amostrais", ylab = "Espécies",
          mar = c(6, 6) + 0.1) # adjust margin size
```



17.10.1 Percentual de encadeamento (% chaining)

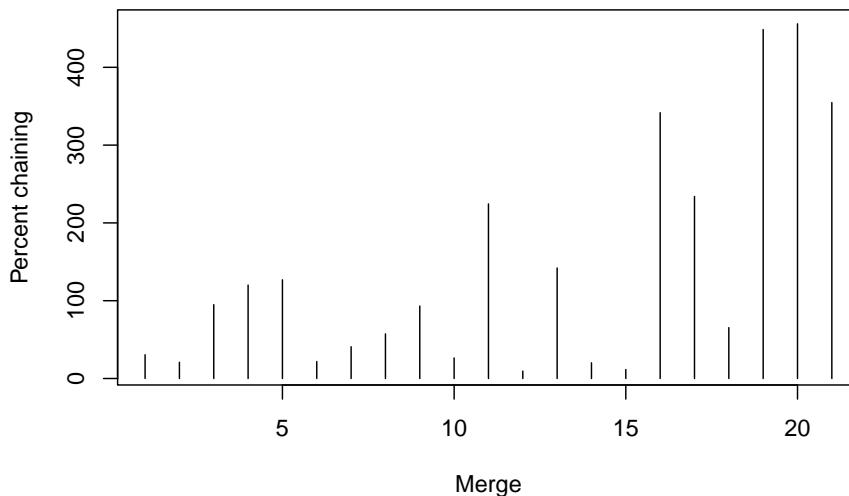
Dendrogram Chaining (ou encadeamento) é um termo usado em análises de cluster hierárquico para se referir ao processo de adição sequencial de pequenos grupos a um ou poucos grandes grupos. É possível usar a porcentagem de encadeamento para ajudar a avaliar a qualidade do dendrograma. Análises altamente encadeadas provavelmente não valem a pena serem exploradas em maiores detalhes, provavelmente há alguma falha nos passos anteriores da análise. Por outro lado, pequenas diferenças na porcentagem de encadeamento na extremidade inferior da escala são geralmente um critério pobre para selecionar um algoritmo de ligação em relação a outro (MCCUNE; GRACE, 2002).

O método para calcular o grau de encadeamento compara o comprimento mé-

dio do caminho do dendrograma com o comprimento mínimo possível (sem encadeamento) e o comprimento máximo possível (encadeamento completo). O comprimento do caminho para cada item no dendrograma é o número de nós que o conectam ao nível mais alto do dendrograma. O encadeamento completo ocorre quando cada fusão envolve a adição de um único item a um único grupo, cada vez mais grande (MCCUNE; GRACE, 2002).

Com esse código podemos calcular o percentual de encadeamento para cada fusão em `cluster1` e fazer um gráfico simples para observação.

```
# calculate the percent chaining for each merge
h <- cluster1$height
n <- length(h)
pc <- rep(NA, n-1)
for (i in 1:(n-1)) {
  pc[i] <- abs(h[i] - h[i+1]) / h[1] * 100
}
# plot the percent chaining
plot(pc, type = "h", xlab = "Merge", ylab = "Percent chaining")
```



Para calcular o percentual geral de encadeamento em um dendrograma, é necessário calcular a diferença na altura entre cada nó e seus nós filhos e, em seguida, somar essas diferenças. Essa soma representa o comprimento total do dendrograma. Em seguida, é necessário calcular a soma das diferenças na altura entre cada nó que representa uma fusão e a altura de seus dois nós filhos.

Essa soma representa o encadeamento total no dendrograma.

Finalmente, você pode calcular o percentual geral de encadeamento como o encadeamento total dividido pelo comprimento total do dendrograma, multiplicado por 100.

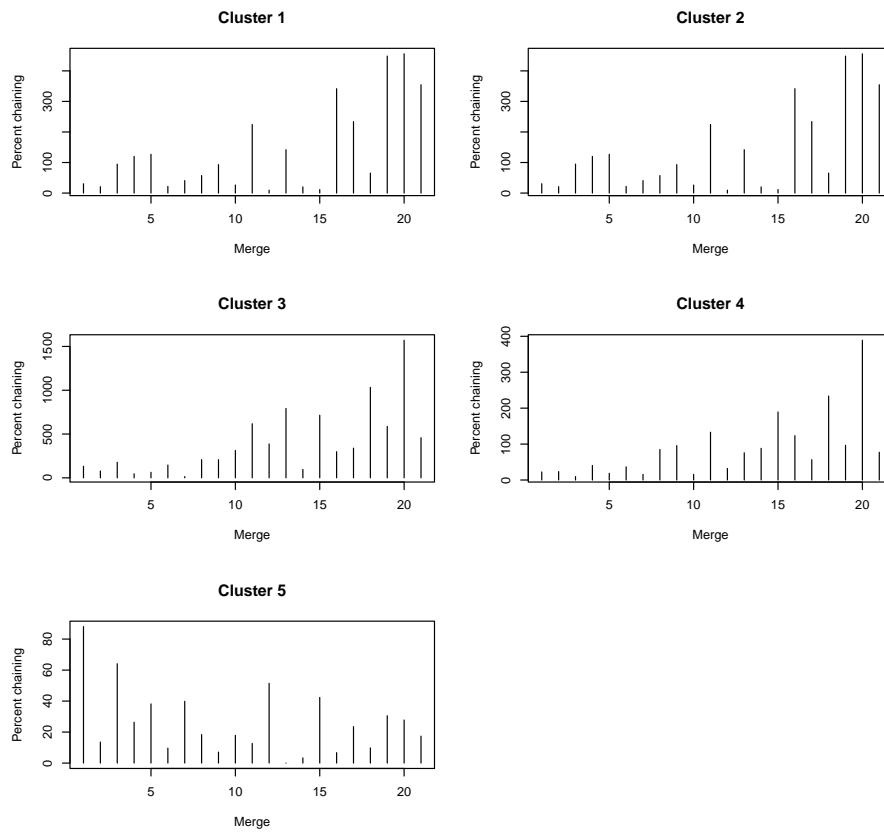
O eixo Y do gráfico mostra a percentagem de encadeamento, que é a diferença na altura entre fusões sucessivas dividida pela altura total do dendrograma, multiplicada por 100. É possível que a percentagem de encadeamento seja superior a 100 se a altura das duas fusões comparadas for maior que a altura total do dendrograma. Nesse caso, a percentagem de encadeamento é truncada em 100.

Isso pode acontecer, por exemplo, se a altura de uma fusão for negativa, o que pode ocorrer se a distância entre dois grupos for menor que zero, ou se o dendrograma for escalado ou truncado de alguma forma.

```
# create a list of the four cluster analyses
cluster_list <- list(cluster1, cluster2, cluster3, cluster4, cluster5)

par(mfrow=c(4,2)) #posicionamento dos gráficos

# loop through the list and calculate percent chaining for each
for (i in 1:5) {
  h <- cluster_list[[i]]$height
  n <- length(h)
  pc <- rep(NA, n-1)
  for (j in 1:(n-1)) {
    pc[j] <- abs(h[j] - h[j+1]) / h[1] * 100
  }
  # plot the percent chaining for each cluster analysis
  plot(pc, type = "h", xlab = "Merge", ylab = "Percent chaining", main = paste0("Clust",
})
```



Apêndices

Sites consultados

<https://copyprogramming.com/howto/how-to-color-labels-of-dendrogram-with-dendextend-and-heatmap-2-using-pre-defined-sample-groups> <https://www.datacamp.com/tutorial/hierarchical-clustering-R>

Outras formas de fazer a partição

```
m_bruta_part2 <- m_bruta_part
```

$$\text{colSums}(\text{abs}(m_{bruta_part}), \text{na.rm} = F) \quad 0$$

```
m_bruta_part2 <- subset(m_bruta_part, colSums != 0) m_bruta_part2 <-
m_bruta_part
, colSums(m_{bruta\_part}! = 0) \quad 0
```

Código

```
range(euclidma) par(mfrow=c(2,1)) hist(euclidma, breaks = 15, #determina o no. de colunas do histograma xlim = range(floor(min(euclidma)), ceiling(max(euclidma))), xlab = "Distr. de Frequências", freq = FALSE) curve(dnorm(x, mean=mean(euclidma), sd=sd(euclidma)), add=TRUE) boxplot.default(euclidma, horizontal = TRUE, frame = FALSE, xlab="Distr. de Frequências", ylim=c(floor(min(euclidma)), ceiling(max(euclidma))))
```

Script limpo

Aqui apresento o scrip na íntegra sem os textos ou outros comentários. Você pode copiar e colar no R para executá-lo. Lembre de remover os `#` ou `##` caso necessite executar essas linhas.

```
## dev.off() #apaga os graficos, se houver algum
## rm(list=ls(all=TRUE)) #limpa a memória
## cat("\014") #limpa o console
## install.packages("openxlsx") #importa arquivos do excel
## install.packages("moments") #calcula assimetria e curtose dos dados
## install.packages("matrixStats") #fornece funções rápidas para a estatística de matrizes
## install.packages("gt") #ferramenta para criação de tabelas bonitas e personalizáveis
## getwd()
## setwd("C:/Seu/Diretório/De/Trabalho")
library(openxlsx)
m_bruta <- read.xlsx("D:/Elvio/OneDrive/Disciplinas/_EcoNumerica/5.Matrizes/peixes06.xlsx",
                      rowNames = T, colNames = T,
                      sheet = "Sheet1")
str(m_bruta)
m_bruta_ma <- as.matrix(m_bruta) #lê m_bruta como uma matriz
str(m_bruta_ma)
#m_bruta
m_bruta_ma[1:5,1:5] #[1:5,1:5] mostra apenas as linhas e colunas de 1 a 5.
#m_bruta <- (m_bruta) # <1>
#View(m_bruta)
print(m_bruta)
```

```

head(m_bruta)
str(m_bruta)
mode(m_bruta)
class(m_bruta)
#?str
range(m_bruta) #menor e maior valores
length(m_bruta) #no. de colunas
ncol(m_bruta) #no. de N colunas
nrow(m_bruta) #no. de M linhas
sum(lengths(m_bruta)) #soma os nos. de colunas
length(as.matrix(m_bruta)) #tamanho da matriz m x n
sum(m_bruta == 0) # número de observações igual a zero
sum(m_bruta > 0) # número de observações maiores que zero
zeros <- (sum(m_bruta == 0)/length(as.matrix(m_bruta)))*100 # proporção de zeros na matriz
zeros
tamanho <- data.frame(
  Função = c("range", "length", "m cols", "n linhas", "Tamanho", "Tamanho",
            "Zeros", "Nao zeros", "% Zeros"),
  Resultado = c(paste(range(m_bruta), collapse = " - "), length(m_bruta), ncol(m_bruta),
                nrow(m_bruta), sum(lengths(m_bruta)), length(as.matrix(m_bruta)), sum(m_bruta ==
                  sum(m_bruta > 0), round(zeros, 1)))
)
tamanho
knitr::kable(tamanho, format = "markdown", caption = "Resumo das informações sobre o tamanho da matriz")
euclid <- dist(m_bruta, method = "euclidean", diag = TRUE, upper = FALSE)
#?dist
#euclid
str(euclid)
mode(euclid)
class(euclid)
length(as.matrix(euclid))
as.matrix(euclid)[1:6, 1:6] #mostra as 5 primeiras linhas e colunas da matriz
euclid_ma <- (as.matrix(euclid))
#View(euclid_ma)
str(euclid_ma)
mode(euclid_ma)
class(euclid_ma)
euclid_ma[1:5, 1:5] #mostra as 5 primeiras linhas e colunas da matriz
range(m_bruta)
range(euclid)
min(euclid)
max(euclid)
mean(euclid) #CENTROIDE!! OU Grand mean
sd(euclid) #Standard deviation

```

```

centroide <- mean(euclid)
centroide
length(euclid)
m <- nrow(m_bruta)
m
m*(m-1)/2
summary(euclid)
Sumario1 <- cbind(min(euclid),
                     max(euclid),
                     sd(euclid),
                     mean(euclid),
                     length(euclid))
colnames(Sumario1) <- c("Minimo", "Maximo", "Desv.Padr", "Media", "m(m-1)/2")
rownames(Sumario1) <- ("Valores")
Sumario1
range(euclid)
par(mfrow=c(2,1))
hist(euclid,
      breaks = 15, #determina o no. de colunas do histograma
      xlim = range(floor(min(euclid)), ceiling(max(euclid))),
      xlab = "Distr. de Frequências",
      freq = FALSE)
curve(dnorm(x, mean=mean(euclid), sd=sd(euclid)), add=TRUE)
boxplot.default(euclid, horizontal = TRUE, frame = FALSE,
                xlab="Distr. de Frequências",
                ylim=c(floor(min(euclid)), ceiling(max(euclid)))) #Limites do eixo Y
## dev.off()
library(matrixStats)
library(moments)
euclid_ma <- as.matrix(euclid)
euclid_ma
range(euclid_ma) #valor errado
mean(euclid_ma) #valor errado
sd(euclid_ma) #valor errado
is.na(euclid_ma) <- euclid_ma==0 #atribui n.a. aos valores = 0
mean(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
mean(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
sd(euclid_ma, na.rm=T) #valor correto, omite valores n.a. do calculo
#colMeans(euclid_ma, na.rm=T) #omite valores n.a. do calculo
#rowMeans(euclid_ma, na.rm=T) #omite valores n.a. do calculo
centroide_ma <- mean(euclid_ma, na.rm=T)
av.dist <- (as.matrix(colMeans(euclid_ma, na.rm=T)))
av.desvpad <- (as.matrix(colSds(euclid_ma, na.rm=T)))
dp.centroide_ma <- (av.dist-centroide_ma)/(colSds(av.dist)) #ou z-scores

```

```

list <- as.matrix(cbind(av.dist, av.desvpad, dp.centroide_ma))
list
colnames(list, do.NULL = FALSE)
colnames(list) <- c("Av.Dist", "Av.StDev", "DP.Centroide")
list2 <- list[order(list[,1], decreasing = TRUE),] #[,1] ou o nome da coluna
list2
par(mfrow=c(3,1))
hist(list2[, "Av.Dist"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias (em dp) para o centroide",
      main = "Distribuição de Frequência da distância média para o centroide",
      xlim = range(floor(min(av.dist)), ceiling(max(av.dist))), #substitua aqui o menor e maior val
      freq = T)
hist(list2[, "Av.Dist"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias (em dp) para o centroide",
      main = "Curva de normalidade ajustada para a Distribuição de Frequência",
      xlim = range(floor(min(av.dist)), ceiling(max(av.dist))), #substitua aqui o menor e maior val
      freq = F)
curve(dnorm(x, mean=mean(list2[, "Av.Dist"])), sd=sd(list2[, "Av.Dist"])), add=TRUE)
boxplot.default(list2[, "Av.Dist"], horizontal = TRUE, frame = FALSE,
               xlab="Distr. de Frequências",
               ylim=c(floor(min(av.dist)), ceiling(max(av.dist)))) #substitua aqui o menor e ma
par(mfrow=c(1,1))
range(dp.centroide_ma)
par(mfrow=c(3,1))
hist(list2[, "DP.Centroide"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias dos desvios padões para o centroide",
      main = "Distribuição de Frequência dos desvio padões das distâncias médias para o centroide",
      xlim = range(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma))), #substitua aqui o
      freq = T)
hist(list2[, "DP.Centroide"],
      breaks = 15, #determina o no. de colunas do histograma
      xlab = "Distr. de Frequências das Distâncias dos desvios padrões das distâncias médias para
      main = "Curva de normalidade ajustada para a Distribuição de Frequência",
      xlim = range(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma))), #substitua aqui o
      freq = F)
curve(dnorm(x, mean=mean(list2[, "DP.Centroide"])), sd=sd(list2[, "DP.Centroide"])), add=TRUE)
boxplot.default(list2[, "DP.Centroide"], horizontal = TRUE, frame = FALSE,
               xlab="Distr. de Frequências",
               ylim=c(floor(min(dp.centroide_ma)), ceiling(max(dp.centroide_ma)))) #substitua a
par(mfrow=c(1,1))
## dev.off()

```


Referências

- BEASLEY, C. R. **Bioestatística usando R. Apostila de exemplos para o biólogo.** [s.l: s.n.]. Disponível em: <<http://www.ead.unicamp.br/minicurso/bw/texto/fdl.pt.html>>.
- BORCARD, D.; GILLET, F.; LEGENDRE, P. **Numerical Ecology with R.** 2. ed. [s.l.] : Springer, 2018. p. 435
- BRADSTOCK, R. A.; TOZER, M. G.; KEITH, D. A. Effects of high frequency fire on floristic composition and abundance in a fire-prone heathland near Sydney. **Australian Journal of Botany**, [s. l.], v. 45, p. 641–655, 1997.
- DIXON, P. VEGAN, a package of R functions for community ecology. **Journal of Vegetation Science**, [s. l.], v. 14, p. 927–930, 2003.
- FERREIRA, E. B.; OLIVEIRA, M. S. de. **Introdução à Estatística com R.** Alfenas, MG: Editora Universidade Federal de Alfenas, 2020. p. 194 Disponível em: <https://www.unifal-mg.edu.br/bibliotecas/wp-content/uploads/sites/125/2021/12/32-EBR_Unifal.pdf>
- GOTELLI, N. J.; ELLISON, A. M. **Princípios de Estatística em Ecologia.** Porto Alegre: Artmed, 2011. p. 527
- HORTON, N. J.; KLEINMAN, K. **Using R and RStudio for Data Management, Statistical Analysis, and Graphics.** 2. ed. [s.l.] : CRC Press, 2015. p. 253
- HURLBERT, S. H. The nonconcept of species diversity: a critique and alternative parameters. **Ecology**, [s. l.], v. 52, n. 4, p. 577–586, 1971.
- LEGENDRE, P.; LEGENDRE, L. **Numerical Ecology.** 2nd English ed ed. Amsterdam, The Netherlands ; New York ; Oxford. p. 853
- LUDWIG, J. A.; REYNOLDS, J. F. **Statistical Ecology. A Primer on Methods and Computing.** New York, NY: John Wiley & Sons, 1988. p. 337
- MAGURRAN, A. E. **Ecological Diversity and Its Measurement.** [s.l.] : Croom Helm Ltd, 1988. p. 179
- MCCUNE, B.; GRACE, J. B. **Analysis of Ecological Communities.** Glendale Beach, Oregon, U.S.A.: MjM Software Design, 2002. p. 300
- MEDEIROS, E. S. F.; ROSA, I. L. Reprodução de tucunaré Cichla ocellaris (Perciformes: Cichlidae) Bloch & Schneider, 1801 na represa do rio Gramame, Alhandra / PB. **Resumos. II Encontro de Iniciação Científica da UFPB,** [s. l.], p. 82–82, 1994.
- MEDEIROS, E. S. F.; SILVA, M. J. da; RAMOS, T. P. A.; RAMOS, R. T. C. Environmental variables as predictors of fish community composition in semiarid aquatic systems. **Acta Limnologica Brasiliensis**, [s. l.], v. 36, n. e4, p. 1–13, 2024.

422CHAPTER 17. R MÓDULO 7 - ANÁLISE DE CLASSIFICAÇÃO HIERÁRQUICA - SAHN

MEDEIROS, E. S. F.; SILVA, M. J.; RAMOS, R. T. C. Application of catchment- and local-scale variables for aquatic habitat characterization and assessment in the Brazilian semi-arid region. **Neotropical Biology and Conservation**, [s. l.], v. 3, n. 1, p. 13–20, 2008. Disponível em: <<https://revistas.unisinos.br/index.php/neotropical/article/view/5440>>

MELO, A. S. **Conversando com o R usando 57 palavras: Introdução à programação com exemplos em Ecologia**. [s.l: s.n.]. Disponível em: <https://ecoevol.ufg.br/adrimelo/prog/Conversando_com_o_R_usando_57_palavras-v13.pdf>.

OKSANEN, J.; BLANCHET, F. G.; FRIENDLY, M.; KINTDT, R.; LEGENDRE, P.; MCGLINN, D.; MINCHIN, P. R.; O'HARA, R. B.; SIMPSON, G. L.; SOLYMOS, P.; STEVENS, M. H. H.; SZOEC, E.; WAGNER, H. **Vegan: Community Ecology Package. R package version 2.5-7**. [s.l.] : The Comprehensive R Archive Network Archive, 2020. Disponível em: <<https://cran.r-project.org/web/packages/vegan/vegan.pdf>>.

OLIVEIRA, P. F. de; GUERRA, S.; McDONNELL, R. **Ciência de dados com R: Introdução**. Brasília, DF: Editora IBPAD, 2018. p. 240 Disponível em: <<https://cdr.ibpad.com.br/>>

PIELOU, E. C. **Ecological Diversity**. New York, NY: John Wiley & Sons, 1975. p. 165

SILVA, E. P. da; DUARTE, M. R. N.; MEDEIROS, E. S. F. Length-weight relationship of two fish species from a dryland intermittent river in northeastern Brazil. **Neotropical Biology and Conservation**, [s. l.], v. 13, n. 1, 2018.

SILVA, F. R. da; GONÇALVES-SOUZA, T.; PATERNO, G. B.; PROVETE, D. B.; VANCINE, M. H. **Análises ecológicas no R**. Recife, PE : Bauru, SP: Nupeea, Canal 6, 2022. Disponível em: <<https://analises-ecologicas.netlify.app/>>

SILVA, M. J.; FIGUEIREDO, B. R. S.; RAMOS, R. T. C.; MEDEIROS, E. S. F. Food resources used by three species of fish in the semi-arid region of Brazil. **Neotropical Ichthyology**, [s. l.], v. 8, n. 4, p. 825–833, 2010.

SOKAL, R. R.; ROHLF, F. J. **Introduction to Biostatistics**. 2. ed. Mineola, New York: Dover Publications, 1987. p. 363

SOKAL, R. R.; ROHLF, F. J. **Biometry: The Principles and Practice of Statistics in Biological Research**. 3. ed. New York: W.H. Freeman; Company, 1995. p. 776

TEAM, R. D. C. **An Introduction to R**. [s.l: s.n.]. Disponível em: <<https://cran.r-project.org/doc/manuals/r-release/R-intro.html>>.

TEAM, R. D. C. **R: A language and environment for statistical computing**. Austria: R Foundation for Statistical Computing, 2017. Disponível em: <<https://www.r-project.org/>>

TEAM, R. S. **RStudio: Integrated Development Environment for R.** Boston, MA: RStudio, PBC, 2022. Disponível em: <<https://posit.co/products/open-source/rstudio/>>

VALENTIN, J. L. **Ecologia Numérica. Uma Introdução à Análise Multivariada de Dados Ecológicos.** 1. ed. Rio de Janeiro: Interciência, 2000. p. 117

WILLIAMS, D. G. **Study Guide: Classification and Ordination.** (v1.1), 2002. ed. Canberra, Australia: Flexible Delivery Development Unit, Centre for Enhancement of Learning, Teaching; Scholarship (CELTS), University of Canberra, Australia, 2002.

ZAR, J. H. **Biostatistical Analysis.** 4th. ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 1999. p. 663