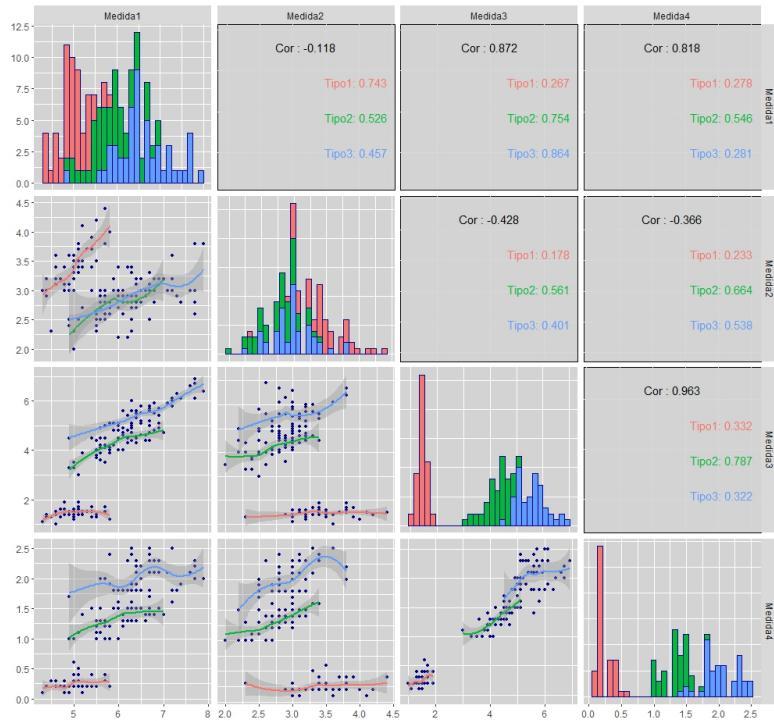


Luciane Ferreira Alcoforado
Ariel Levy

Visualização de dados com o software R



Luciane Ferreira Alcoforado

Ariel Levy

Visualização de dados com o software R

1^a. Edição

Niterói-RJ

LFA

2017

Universidade Federal Fluminense

DATAUFF – Núcleo de Pesquisas, Informações e Políticas Públicas

Diretora: Luciane Ferreira Alcoforado

www.uff.br/datauff

R. Passo da Pátria, 156 São Domingos, Niterói – RJ CEP – 242210-240

TEL. +55 21 2629-5410

Capa: Ariel Levy e Luciane F. Alcoforado

Edição de Texto: Luciane F. Alcoforado

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da
Universidade Federal Fluminense

A354 Alcoforado, Luciane Ferreira
Visualização de dados com software R / Luciane Ferreira
Alcoforado, Ariel Levy. – Niterói, RJ : LFA, 2017.

Livro eletrônico

1. Software livre. 2. Computação estatística. I. Levy, Ariel. II.
Título.

CDD 005.3

ISBN: 978-85-922932-0-8

COMITÊ EDITORIAL

Manuel Frbrero Bande - USC/ES

Luciane Ferreira Alcoforado - UFF

Orlando Celso Longo - UFF

Ariel Levy - UFF

Emil de Souza Sanchez Filho - UFF

Carlos Alberto Pereira Soares - UFF

Assed Naked Haddad - UFRJ

Maysa Sacramento de Magalhães - ENCE/IBGE

José Rodrigo de Moraes - UFF

Steven Dutt Ross - UNIRIO

Djalma Galvão Carneiro Pessoa - ENCE/IBGE

Pedro Costa Ferreira - FGV/IBRE

Jorge Passamani Zubelli - IMPA

It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment.

Carl Friedrich Gauss

Prefácio

O aumento espantoso da geração de dados em formato digital ocorrido nas últimas décadas levou as organizações a uma nova era com relação à maneira como tomam suas decisões, definem suas políticas e conduzem seus negócios. Chamamos esta era, que está apenas começando, de ERA ANALÍTICA.

Nos próximos anos, as organizações (governos, ONGs, empresas etc.) terão que aprender a fazer uso dos dados gerados por elas próprias, por seus clientes, por seus pares e até mesmo pelos seus concorrentes, sob a pena de não sobreviverem à nova realidade da competitividade e eficiência.

O que permite que esta revolução ocorra é o avanço no poder computacional associado à drástica redução dos custos envolvidos. Chegamos ao ponto de tornar possível o uso em máquinas comuns de métodos analíticos sofisticados, que antes estavam restritos a organizações poderosas e capazes de ter acesso a computadores caríssimos.

Para que todo esse poder seja utilizado, os softwares também tiveram que evoluir, de modo a oferecer ferramentas de tratamento, armazenamento e análise de grandes quantidades de dados e o uso de métodos analíticos sofisticados. E foi então que um fenômeno espetacular aconteceu. A comunidade mundial de desenvolvedores de softwares e analistas de dados abraçou um software interessante, mas limitada que era utilizada quase que apenas por acadêmicos e aficionados em estatística. Este software se chama R e se tornou o centro dessa grande revolução ao passar a oferecer recursos poderosos, interfaces amigáveis e desempenho surpreendente em software totalmente gratuito!

O desafio então passou a ser aprender a utilizar o R de maneira eficaz. Aqui no Brasil, ainda existe uma literatura incipiente e este livro vem suprir esta falta, oferecendo a nós,

cientistas de dados, uma fonte organizada, completa e bem estruturada para dominarmos uma das facetas mais fascinantes e complexas da ciência dos dados, a visualização.

O trabalho do cientista de dados começa e termina na visualização de dados. Começa porque o primeiro passo em toda análise exploratória de dados é a visualização dos dados e suas relações em diversas formas diferentes para ajudar o analista a identificar padrões e sugerir caminhos a seguir na busca por significado nos dados analisados. Após o trabalho de análise, o desenvolvimento e ajuste do modelo, o cientista precisa comunicar suas descobertas às pessoas que tomaram decisões baseadas nas análises. Nesse trabalho de comunicação, o uso de técnicas da *storytelling* passa, quase que obrigatoriamente, pela criação de gráficos criativos e significativos, que apresentam de forma clara o resultado do trabalho e suas recomendações.

O R oferece uma variedade enorme de recursos de visualização de dados, desde os recursos nativos mais simples até pacotes com recursos poderosos, que tornam o R a melhor ferramenta para criação de visualizações sofisticadas, complexas e, porque não, muito belas.

Este livro cobre os pacotes mais populares e estáveis, de modo a permitir que o leitor que faça desde gráficos simples até visualizações sofisticadas em painéis e envolvendo diversas variáveis. A abordagem didática dos autores e a linguagem direta e clara facilita a rápida compreensão ao mesmo tempo que permite o aprofundamento necessário ao cientista mais exigente que visa o desenvolvimento de visualizações sofisticadas.

Espero que vocês aproveitem o livro assim como eu.

Dr. Eduardo Camilo da Silva, D.Sc. CAP®
Cientista de Dados e Apaixonado pelo R

Apresentação

Esta obra é fruto de pesquisas sobre as possibilidades de elaboração de gráficos no mundo atual. As tecnologias disponíveis tornam possível analisar em curto espaço de tempo uma grande massa de dados. Entretanto é preciso ter o suporte de um *software* que possibilite a operacionalização destas análises. Apresentamos aqui o que há de mais atual e inovador no que diz respeito ao uso do *software R* para produção de gráficos e visualização de dados.

O capítulo 2 foi desenvolvido a partir de material produzido durante o projeto de monitoria da disciplina de Métodos Computacionais I no ano de 2011 tendo como monitora a aluna do curso de Graduação em Estatística Paola da Silva Martins, época em que ministramos uma oficina de gráficos durante a agenda acadêmica 2011. O resultado desta experiência e aprendizado gerou a ideia inicial desta obra.

O capítulo 3 aborda o pacote *Lattice* onde o R evolui em relação ao pacote básico nos gráficos condicionados a uma ou mais variável categórica. Apresenta-se além desta característica o efeito *jitter*, o uso de transparência nos gráficos e o uso de gráficos em 3D.

O capítulo 4 apresenta os recursos gráficos do pacote *ggplot2*, que é um dos mais importantes atualmente para visualização dos dados. Ele veio para aprimorar a junção entre o que é possível realizar nos pacotes básico e *Lattice*, de forma extremamente alinhada.

O capítulo 5 apresenta um importante recurso para produção de infográficos interativos usando o pacote *googleVis*, possibilitando ao leitor produzir gráficos para serem publicados na web.

No capítulo 6 aborda-se a utilização dos dispositivos de saída para as formas gráficas, as vantagens e desvantagens de cada formato e o uso das cores para evidenciar os aspectos principais de um gráfico.

Procurou-se estabelecer uma diretriz quanto à escolha do gráfico mais adequado à variável de interesse. Explica-se como inserir cada elemento necessário a um gráfico informativo como título, legenda, nomes nos eixos, escala dos eixos, aplicação de cores, tamanho de fonte, dentro outros objetos gráficos. Apresentamos uma seção de treinamento em alguns capítulos permitindo ao leitor o desenvolvimento da habilidade de manipulação dos comandos aqui apresentados.

A sequência de comandos pode ser adaptada para situações semelhantes em que seja necessária a construção de um dos gráficos apresentados. Pode ser útil a todos aqueles que desejam aprender a construir gráficos adequados utilizando o programa R de forma prática e rápida. Supõe-se alguma familiaridade no R no que diz respeito à criação de objetos do tipo vetor, matriz e tabela de dados.

Esperamos que este trabalho possa inspirar muitos profissionais a produzirem informações relevantes através das inúmeras formas de visualização de dados que o *software* R proporciona.

Os autores

Sumário

1	Surgimento do R.....	14
1.1	Na UFF	14
1.2	No Brasil	15
1.3	No Mundo	16
1.4	Instalando o R	16
2	Gráficos usando pacote básico do R	19
2.1	Variáveis Qualitativas:.....	19
2.2	Variáveis Quantitativas:.....	19
2.3	Os gráficos	20
2.4	Gráfico de barras:.....	20
2.4.1	Modificando o tamanho da fonte.....	22
2.4.2	Eliminando os eixos do gráfico	23
2.4.3	Invertendo a posição das barras.....	24
2.4.4	Cor e sombreamento das barras.....	24
2.4.5	Eliminando a linha de borda das barras.....	28
2.4.6	Gráfico de Barras com duas variáveis	29
2.4.7	Hora de Praticar	33
2.5	Gráfico de setor.....	35
2.5.1	Hora de Praticar	38
2.6	Histograma.....	40
2.6.1	Hora de Praticar	43
2.7	Boxplot.....	44
2.7.1	Hora de Praticar	48
2.8	Gráfico de dispersão	49
2.8.1	Hora de Praticar	52

2.9	Gráfico de linhas	54
2.9.1	Hora de Praticar	58
2.10	Gráfico da distribuição normal	59
2.10.1	Hora de Praticar	63
2.11	Referência Bibliográfica do capítulo 2	65
3	Gráficos usando pacote Lattice	66
3.1	Gráfico de Dispersão no Lattice	67
3.2	Gráfico Stripplot	70
3.3	Gráfico em 3D	71
3.4	Referências Bibliográficas do Capítulo 3	72
4	Gráficos usando pacote ggplot2	73
4.1	Gráfico de Barras no ggplot2	74
4.1.1	Hora de Praticar	78
4.2	Gráfico de Setor no ggplot2	80
4.3	Histograma no ggplot2	80
4.4	BoxPlot no ggplot2	82
4.4.1	Hora de Praticar	86
4.5	Gráfico de dispersão no ggplot2	87
4.6	Gráfico de linhas no ggplot2	89
4.6.1	Hora de Praticar	91
4.7	Títulos dos gráficos	91
4.8	Referência Bibliográfica do capítulo 4	92
5	Gráficos usando pacote googleVis	93
5.1	Instalando o pacote	93
5.2	Os gráficos de movimento (Motion Chart) do Google	94
5.3	Gráficos de Linha do Google	98

5.3.1	Hora de Praticar	99
5.4	Gráficos de Setores do Google	101
5.4.1	Hora de Praticar	101
5.5	Gauge	101
5.5.1	Hora de Praticar	102
5.6	Gráficos Geográficos do Google	103
5.6.1	Hora de Praticar	107
5.7	Referência Bibliográfica do capítulo 5	108
6	Dispositivos Gráficos e Cores no R	109
6.1	Utilizando Múltiplos Dispositivos	111
6.2	Gráficos e suas Cores no R	112
6.3	Referência Bibliográfica do capítulo 6	114
8	Apêndice A	115
9	Apêndice B	119
10	Apêndice C	123

LISTA DE FIGURAS E TABELAS

FIGURA 1-1: LOGO DO R, VERSÃO ANTIGA E ATUAL NA SEQUÊNCIA DA ESQUERDA PARA DIREITA.....	14
FIGURA 2-1: GRÁFICO DE BARRAS RESULTADO DO COMANDO BARPLOT	21
FIGURA 2-2: GRÁFICO DE BARRAS RESULTADO DO COMANDO BARPLOT COM SUBTÍTULO.....	22
FIGURA 2-3: GRÁFICO DE BARRAS SEM OS EIXOS XY	24
FIGURA 2-4: GRÁFICO DE BARRAS NA POSIÇÃO HORIZONTAL	24
FIGURA 2-5: GRÁFICO DE BARRAS COM PREENCHIMENTO DE LINHAS INCLINADAS	25
FIGURA 2-6: GRÁFICO DE BARRAS COM PREENCHIMENTO DE LINHAS INCLINADAS À 90 GRAUS.....	26
FIGURA 2-7: GRÁFICO DE BARRAS COM PREENCHIMENTO DE COR ATRAVÉS DAS OPÇÕES DE COLORS().....	27
FIGURA 2-8: GRÁFICO DE BARRAS COM PREENCHIMENTO DE COR ATRAVÉS DA ATRIBUIÇÃO DE NÚMEROS....	27
FIGURA 2-9: GRÁFICO DE BARRAS COM PREENCHIMENTO DE COR ATRAVÉS DO SISTEMA RGB.....	28
FIGURA 2-10: GRÁFICO DE BARRAS COM PREENCHIMENTO DE COR E SEM BORDA NAS BARRAS.....	28
FIGURA 2-11: GRÁFICO DE BARRAS COM PREENCHIMENTO DE LINHAS E BORDA NAS BARRAS NO MESMO ESTILO.....	29
FIGURA 2-12: GRÁFICO DE BARRAS COM DUAS VARIÁVEIS: INSTITUIÇÃO A E B, BARRAS EMPILHADAS.....	30
FIGURA 2-13: GRÁFICO DE BARRAS COM DUAS VARIÁVEIS: INSTITUIÇÃO A E B, BARRAS LADO A LADO.....	31
FIGURA 2-14: GRÁFICO DE BARRAS COM DUAS VARIÁVEIS: INTITUIÇÃO A E B, BARRAS LADO A LADO COM LEGENDA	32
FIGURA 2-15: GRÁFICO DE SETORES PARA OS DADOS DA FROTA DE VEÍCULOS DE NITERÓI EM 2009 SEM ACABAMENTO.....	36
FIGURA 2-16: GRÁFICO DE SETORES PARA OS DADOS DA FROTA DE VEÍCULOS DE NITERÓI EM 2009 COM TÍTULO E LEGENDA	37
FIGURA 2-17: GRÁFICO DE SETORES PARA OS DADOS DA FROTA DE VEÍCULOS DE NITERÓI EM 2009 COM TÍTULO E RÓTULOS DE PORCENTAGEM	37
FIGURA 2-18: HISTOGRAMA DO CONSUMO DE MILHAS POR GALÃO	41
FIGURA 2-19: HISTOGRAMA DA FREQUÊNCIA RELATIVA DO CONSUMO DE MILHAS POR GALÃO.....	41
FIGURA 2-20: HISTOGRAMA COM LINHAS DE SOMBREAMENTO	41
FIGURA 2-21: HISTOGRAMA COM TÍTULO PRINCIPAL E RÓTULOS NOS EIXOS.	42
FIGURA 2-22: BOX PLOT ILUSTRATIVO.	44
FIGURA 2-23: BOX PLOT COM TÍTULO	45
FIGURA 2-24: BOX PLOT PARA VÁRIOS GRUPOS	46
FIGURA 2-25: BOX PLOT HORIZONTAL	47
FIGURA 2-26: DIAGRAMA DE DISPERSÃO E A RETA DE REGRESSÃO.....	50
FIGURA 2-27: DIAGRAMA DE DISPERSÃO VÁRIOS LADO A LADO	52
FIGURA 2-28: GRÁFICO DE DUAS FUNÇÕES	54
FIGURA 2-29: GRÁFICO DE LINHA DA SÉRIE DE TEMPERATURA DE ITHACA - NY	55
FIGURA 2-30: GRÁFICO DE LINHA E PONTOS DA SÉRIE DE TEMPERATURA DE ITHACA - NY	56
FIGURA 2-31: GRÁFICO DE LINHA E PONTOS DE DUAS SÉRIES TEMPORAIS COM LEGENDA	57
FIGURA 2-32: GRÁFICO DA DISTRIBUIÇÃO NORMAL DE MÉDIA 0 E DESVIO PADRÃO 1.....	59

FIGURA 2-33: HISTOGRAMA DE UMA VARIÁVEL COM DISTRIBUIÇÃO NORMAL	60
FIGURA 2-34: GRÁFICO DE HISTOGRAMA COM A CURVA DA NORMAL	61
FIGURA 2-35: GRÁFICO DE DUAS NORMAIS COM MÉDIAS DIFERENTES E LEGENDA.....	62
FIGURA 2-36: GRÁFICO DE DUAS NORMAIS COM VARIÂNCIAS DIFERENTES	63
FIGURA 3-1: GRÁFICO DE DISPERSÃO – DADOS MTCARS PACOTE LATTICE	67
FIGURA 3-2: GRÁFICO DE DISPERSÃO CONDICIONADO AO NÚMERO DE CILINDROS.....	68
FIGURA 3-3: GRÁFICO DE DISPERSÃO DUPLAMENTE CONDICIONADO	69
FIGURA 3-4: GRÁFICO DE DISPERSÃO DUPLAMENTE CONDICIONADO COM O PACOTE BÁSICO.	70
FIGURA 3-5: STRIPLOT EXEMPLO DE GRÁFICO DE TIRAS – SARKAR (2015).....	71
FIGURA 3-6: WIREFRAME GRÁFICO EM 3D	72
FIGURA 4-1: GRÁFICOS DE BARRAS LADO A LADO NO GGPLOT	76
FIGURA 4-2: GRÁFICOS DE BARRAS EMPILHADO NO GGPLOT	77
FIGURA 4-3: GRÁFICOS DE BARRAS EMPILHADO NO GGPLOT, FUNDO CINZA, FREQUÊNCIA RELATIVA	78
FIGURA 4-4: GRÁFICOS DE BARRAS EMPILHADO NO GGPLOT, FUNDO BRANCO, FREQUÊNCIA ABSOLUTA.....	78
FIGURA 4-5: GRÁFICOS DE SETORES NO GGPLOT.....	80
FIGURA 4-6: HISTOGRAMA NO GGPLOT	81
FIGURA 4-7: HISTOGRAMA NO GGPLOT COM TERMÔMETRO DE CORES PARA AS CONTAGENS.....	82
FIGURA 4-8: GRÁFICOS BOXPLOT COM ALTERAÇÃO DE COR E INCLUSÃO DE TÍTULOS	83
FIGURA 4-9: GRÁFICOS BOXPLOT COM VÁRIOS GRUPOS	84
FIGURA 4-10: GRÁFICOS BOXPLOT COM VÁRIOS GRUPOS, LEGENDA E CORES	84
FIGURA 4-11: GRÁFICOS BOXPLOT COM VÁRIOS GRUPOS INCLUINDO DISPERSÃO DOS DADOS.....	85
FIGURA 4-12: GRÁFICOS BOXPLOT COM VÁRIOS GRUPOS INCLUINDO DISPERSÃO NA POSIÇÃO HORIZONTAL ..	86
FIGURA 4-13: GRÁFICOS DE DISPERSÃO PARA 5 ESPÉCIES DE ÁRVORES NO GGPLOT	87
FIGURA 4-14: GRÁFICOS DE DISPERSÃO PARA 5 ESPÉCIES DE ÁRVORES COM CORES POR CATEGORIA.....	88
FIGURA 4-15: GRÁFICOS DE DISPERSÃO PARA 5 ESPÉCIES DE ÁRVORES COMANDO JITTER.....	88
FIGURA 4-16: GRÁFICOS DE LINHAS PARA DUAS FUNÇÕES NO GGPLOT	89
FIGURA 4-17: GRÁFICOS DE LINHAS SIMPLES NO GGPLOT	90
FIGURA 4-18: GRÁFICOS DE LINHAS E PONTOS PARA A SÉRIE DE TEMPERATURA DE ITHACA.	90
FIGURA 5-1: GRÁFICOS DE MOVIMENTO – BOLHAS	96
FIGURA 5-2 GRÁFICOS DE MOVIMENTO: BARRAS E LINHAS	97
FIGURA 5-3: EDITOR DE GRÁFICOS DO GOOGLE USANDO O ARGUMENTO “EDIT ME!”.....	97
FIGURA 5-4: EDITOR DE GRÁFICOS DO GOOGLE USANDO O ARGUMENTO “EDIT ME!”.....	99
FIGURA 5-5: EDITOR DE GRÁFICOS DO GOOGLE PERSONALIZAÇÃO	99
FIGURA 5-6: POSSIBILIDADES PARA O GRÁFICO DE SETORES DO GOOGLE.....	101
FIGURA 5-7: GAUGES DO GOOGLE	102
FIGURA 5-8: GEOGRÁFICO PADRÃO DO GOOGLE	103
FIGURA 5-9: GEOGRÁFICO EDITADO NO GOOGLE	104
FIGURA 5-10: GEOGRÁFICO EDITADO NO GOOGLE, REGIÃO ÁSIA	105
FIGURA 5-11: GEOGRÁFICO E TABELA DE DADOS NA MESMA JANELA.....	106

FIGURA 5-12: GEOGRÁFICO COM MARCAS PARA ESPERANÇA DE VIDA DE ALGUNS PAÍSES.....	106
---	-----

TABELA 1.1: ESPERANÇA DE VIDA NO BRASIL	16
TABELA 2.1: NÚMERO DE BOLSAS DE PESQUISA POR ÁREA DE CONHECIMENTO.	20
TABELA 2.2: FROTA DE PRINCIPAIS TIPOS DE VEÍCULO DO MUNICÍPIO DE NITERÓI EM 2009.....	35
TABELA 2.3: TEMPERATURA MÉDIA MENSAL DE ITHACA - NY.....	55
TABELA 3.1: TIPOS DE GRÁFICO NO LATTICE – ADAPTADO DE KOBAKOF (2014) E SAKAR (2015)	66
TABELA 4.1: POSSIBILIDADES DE FORMA PARA GRÁFICOS NO GGPLOT2.....	74
TABELA 4.2: ARQUITETURA DO BANCO DE DADOS PARA GRÁFICO DE BARRAS GGPLOT	74
TABELA 5.1: DADOS SOBRE O ÍNDICE DE APROVAÇÃO DE ALUNOS.....	94
TABELA 5.2: DADOS SOBRE O NÚMERO DE HOMENS (H) E MULHERES (M) EM CADA CURSO	98
TABELA 5.3: DADOS SOBRE ESPERANÇA DE VIDA AO NASCER	103

1 Surgimento do R

Luciane Ferreira Alcoforado

O software R surgiu no final da década de 1990, na Universidade de Auckland, Nova Zelândia através da iniciativa de dois professores de Estatística que perceberam a importância de se ter acesso a um código acessível aos alunos. Sua primeira versão foi publicada no dia 29 de fevereiro de 2000.



Figura 1-1: Logo do R, versão antiga e atual na sequência da esquerda para

Em geral, as licenças para software estatísticos comerciais apresentam um custo elevado, o que gera grande dificuldade de manipulação e aprendizagem aos alunos que necessitam desta ferramenta para produzir suas análises.

Outro aspecto importante de se ressaltar é que os softwares comerciais possuem seu código fechado, não permitindo ao usuário qualquer intervenção, ficando o mesmo sujeito à tarefa de operacionalizar os menus.

Sensíveis a isto, Robert Gentleman e Ross Ihaka iniciaram a primeira versão do R que foi disponibilizada somente para seus alunos. A iniciativa foi tão bem recebida que eles acabaram compartilhando com outros e o resultado foi o software rapidamente disseminado pelo mundo.

O grande sucesso desta iniciativa é que o sistema que se criou de compartilhamento e colaboração resultou na explosão de uso do R no meio acadêmico, sendo absorvido pelas empresas mais inovadoras ao perceberem a grande flexibilidade que seu uso acarreta, além é claro da economia financeira para licença de uso.

1.1 Na UFF

Não sabemos precisar quem foram os pioneiros a utilizar o R dentro da UFF, mas no ano de 2010 seu aprendizado foi introduzido no curso de Estatística na disciplina obrigatória

de Métodos Computacionais I, o primeiro contato do aluno calouro com o R. Naquela época, cem por cento dos alunos ingressantes neste curso desconheciam o que era o R. Ainda hoje a maioria dos calouros não o conhecem, o que indica que é no meio Universitário que se tem este primeiro contato.

“Meu primeiro contato com o software foi em 2006 num curso de atualização que eu realizava na ENCE. Naquela época eu pude perceber a potencialidade do software, uma linguagem voltada a objeto, proporcionando facilidade para quem precisa realizar uma pequena programação de comandos, muito mais fácil do que os programas que exigiam declaração de variáveis, uma compilação e posterior execução.”¹

Atualmente a Universidade Federal Fluminense abriga um importante portal www.estatisticacomr.uff.br que reúne diversos conteúdos de aprendizagem para toda a comunidade, sendo a pioneira no Brasil em organizar um seminário Internacional sobre a linguagem, o SER cuja primeira edição ocorre em maio de 2016.

1.2 No Brasil

No Brasil a porta de entrada do R foi através da Universidade Federal do Paraná (UFPR), que disponibilizava um espelho para realização de *download* do R. As primeiras apostilas em português também iniciaram através da produção do professor Paulo Justiniano da UFPR, (R-BR, 2015). Atualmente há uma lista de discussão oficial do programa R, a R-BR com o propósito de permitir a troca de informações entre os usuários de R no Brasil.

Os canais que se destacam na rede social do facebook, tratando especificamente desta linguagem são o grupo R Brasil Programadores contando com aproximadamente 1.500 membros, o grupo R UNIRIO com 110 membros, a página Estatística usando o R com 1700 curtidas e a página Estatística com R com 350 curtidas, esta última criada em maio de 2015 e administrada por equipe de professores e alunos da UFF. Para se ter uma base de comparação podemos citar a página oficial da comunidade Estatística Brasil que conta com 3100 curtidas, (Facebook, 2015).

¹ Luciane Ferreira Alcoforado em entrevista ao Jornal O Fluminense, Segundo Caderno, 09/09/14.

Podemos perceber através destes números, que assuntos de interesse acadêmico como Estatística e software R não são muito populares se comparados a outros temas nas redes sociais, a maioria destas páginas foram criadas há menos de 4 anos e vem gradativamente ganhando adesão de professores de Instituições de Ensino. Pode-se perceber que muitos grupos e páginas foram criados por iniciativa de alunos de pós-graduação ou graduação no intuito de compartilhar suas dúvidas e promover uma rede de colaboração.

Temos disponível cinco espelhos para *download* do software R:

Tabela 1.1: Espelhos do software R no Brasil

http://nbcgib.uesc.br/mirrors/cran/	Center for Comp. Biol. at Universidade Estadual de Santa Cruz
http://cran-r.c3sl.ufpr.br/	Universidade Federal do Paraná
http://cran.fiocruz.br/	Oswaldo Cruz Foundation, Rio de Janeiro
http://www.vps.fmvz.usp.br/CRAN/	University of São Paulo, São Paulo
http://brieger.esalq.usp.br/CRAN/	University of São Paulo, Piracicaba

Fonte: <https://cran.r-project.org/mirrors.html>

1.3 No Mundo

Vem ganhando crescente popularidade, tanto é que em meados de 2015 o software atingiu a marca da sexta linguagem mais popular do mundo, segundo a métrica da IEEE Spectrum. Na mesma época em que tal notícia foi divulgada também era anunciado o R consortium que reúne Linux, Microsoft e o R time. Este consortium promete trabalhar no apoio ao desenvolvimento do R, mantendo a linguagem e prestando suporte aos seus usuários.

Na rede social há diversas páginas e grupos, das quais destacamos o R bloggers com 28.500 curtidas, o grupo Users of R Statistical Package com 11.500 membros e o grupo Big Data Analytics using R com 6.500 membros, (Facebook, 2015).

1.4 Instalando o R

Para fazer download do programa R, utilize sempre a página oficial do projeto, disponível em www.r-project.org. Nesta página estará disponível diversas informações sobre

a linguagem, documentos que podem auxiliar o aprendiz, além de ser constantemente atualizada com novas versões do programa.

Observe que ao lado direito da página oficial, abaixo da palavra *download*, há um *link* para CRAN que dará acesso aos espelhos do programa distribuídos pelos diversos países do mundo. No Brasil atualmente há 5 opções, qualquer uma delas pode ser utilizada, veja Tabela 1.

O programa é multiplataforma, o que significa que pode ser instalado nos sistemas Windows, Linux e Mac. Assim, na página de *download* opte pelo *link* disponível para o seu sistema operacional. Veja a ilustração para a opção Windows:

Passo 1: Selecione o sistema operacional

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

Passo 2: Selecione a versão disponível (na ilustração é a versão R 3.2.1)

[Download R 3.2.1 for Windows](#) (62 megabytes, 32/64 bit)
[Installation and other instructions](#)
[New features in this version](#)

Esta seleção fará o *download* do programa executável que possibilitará a instalação no seu computador. Atente para o fato de que constantemente são disponibilizadas novas versões do programa e suas atualizações não são automáticas, cabendo ao usuário verificar nesta página a ocorrência de novas versões.



Pronto, agora é só utilizar o programa, clicando no ícone

Nos próximos capítulos abordaremos as possibilidade de visualização de dados com o R, entretanto gostaríamos de destacar que a linguagem oferece inúmeras outras.

Referências Bibliográficas – Capítulo 1

- Alcoforado, L.F. e equipe. **Portal Estatística é com R!** Disponível em www.estatisticacomr.uff.br. Acesso em 08/12/15.
- Facebook. **Rede Social**. Disponível em www.facebook.com. Acesso em 08/12/15.
- R-BR. **Lista Brasileira Oficial de discussão do programa R**. Universidade Federal do Paraná – LEG. Disponível em <http://www.leg.ufpr.br/doku.php/software:rbr>. Acesso em 08/12/15.
- R Development Core Team. **R: a language and environment for statistical computing**. Viena (A): R Foundation for Statistical Computing. 2009. Disponível em <http://www.r-project.org>. Acesso 08/12/15.
- Ribeiro Jr, P.J. Brown, P.E. **Some Words on the R project**. The ISBA Bulletinm, v.8, n.1, p. 1216, 2001.
- Rbloggers. **R news and tutorials contributed**. Disponível em <http://www.rbloggers.com>. Acesso em 08/12/15.

2 Gráficos usando pacote básico do R

Luciane Ferreira Alcoforado

2.1 Variáveis Qualitativas:

Características que dizem respeito a qualidade ou atributos. Se as categorias da variável possuem uma ordem natural, ela é dita ser *qualitativa ordinal*; caso contrário, *qualitativa nominal*. No R estas variáveis são chamadas de fatores (*factor*).

Exemplos:

- *Variáveis Qualitativas Nominais:*

Sexo: Feminino, Masculino

Cor de olhos: Pretos, Castanhos, Azuis, Verdes

Tipo de Farinha: Trigo, aveia, milho, banana, outra

Fator RH do sangue: positivo, negativo

- *Variáveis Qualitativas Ordinais:*

Estado de Saúde: Ruim, Regular, Bom

Tipo de Acidente: Leve, Moderado, Grave

Cargo na empresa: diretor, vice-presidente, presidente

Quais gráficos podem ser usados?

Setores e barras

2.2 Variáveis Quantitativas:

Características que dizem respeito a quantidades. Se os valores da variável se referem a uma contagem, ela é dita ser *quantitativa discreta*. Caso se refira a uma mensuração (com unidade de medida), ela é dita ser *quantitativa contínua*.

Exemplos:

- *Variáveis Quantitativas Discretas:*

Nº de pessoas na família

Nº de acidentes na BR101 por ano

Nº de anos de estudos

- *Variáveis Quantitativas Contínuas:*

Peso (Kg)

Valor de passagem aérea (R\$)

Tempo para executar uma tarefa (horas)

Quais gráficos podem ser usados?

Histograma, boxplot, dispersão, linha.

2.3 Os gráficos

A elaboração de gráficos no R se dará na forma de *script*. Construiremos os *scripts* que poderão ser adaptados para outras situações semelhantes. Para executar os *scripts* basta inseri-los no console do programa R.

2.4 Gráfico de barras:

O gráfico de barras é composto por duas linhas ou eixos, um vertical e outro horizontal. No eixo vertical são construídas as barras que representam a variação de um fenômeno ou de um processo de acordo com sua intensidade. Essa intensidade é indicada pela altura da barra. No eixo horizontal especifica-se as categorias da variável. As barras devem sempre possuir a mesma largura e a distância entre elas deve ser constante.

Veja a seguir como construir este tipo de gráfico no R. Utilizaremos os dados da tabela 3.1 referente ao número de bolsas de pesquisa em cada área do conhecimento de duas instituições de ensino:

Tabela 2.1: Número de bolsas de pesquisa por área de conhecimento.

Área do conhecimento	Instituição A	Instituição B
Saúde	1751	2528
Exatas	2186	2132
Humanas	947	1843
Comunicação	29	280

Fonte:Dados fictícios

O primeiro passo na construção do gráfico é ter os dados armazenados em objeto apropriado. No caso de gráfico de barras é necessário que os dados da tabela estejam armazenados em vetores que denominaremos de x e y , sendo que x representará as categorias das áreas do conhecimento e y representará as frequências de cada categoria. Como temos duas instituições, discriminaremos as frequências por ya e yb respectivamente a cada instituição.

Entretanto, o vetor yb não terá nenhuma função nos *scripts* em que consideram apenas a instituição A e pode ser omitido até que seja necessário a sua utilização.

Script Gráfico de Barras sem subtítulo

```
x<-c("saúde", "exatas", "humanas", "comunicação")
ya<-c(1751, 2186, 947, 29)
yb<-c(2528, 2132, 1843, 280)
barplot(ya, names. arg=x, main= "Distribuição de bolsas de pesquisa por área de
conhecimento", xlab= "Grandes Áreas", ylab="Frequência")
```

Note que o título do gráfico é especificado pelo texto entre aspas do argumento `main` do comando `barplot`, enquanto que os eixos são nomeados pelo argumento `xlab` e `ylab` respectivamente ao eixo x e y . O script acima produzirá o gráfico da Figura 2-1:

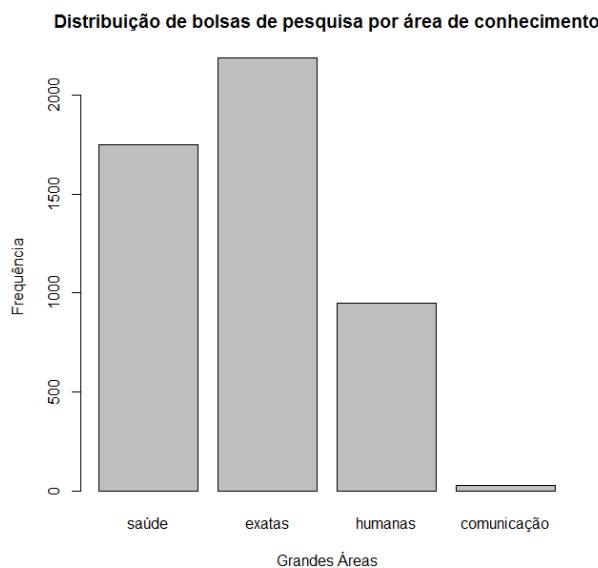


Figura 2-1: Gráfico de barras resultado do comando barplot

Caso deseje acrescentar um subtítulo que ficará posicionado abaixo do eixo x, o script deverá conter o argumento `sub="texto"`, veja:

```

Script Gráfico de Barras com subtítulo

x<-c("saúde", "exatas", "humanas", "comunicação")
ya<-c(1751, 2186, 947, 29)
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios")
```

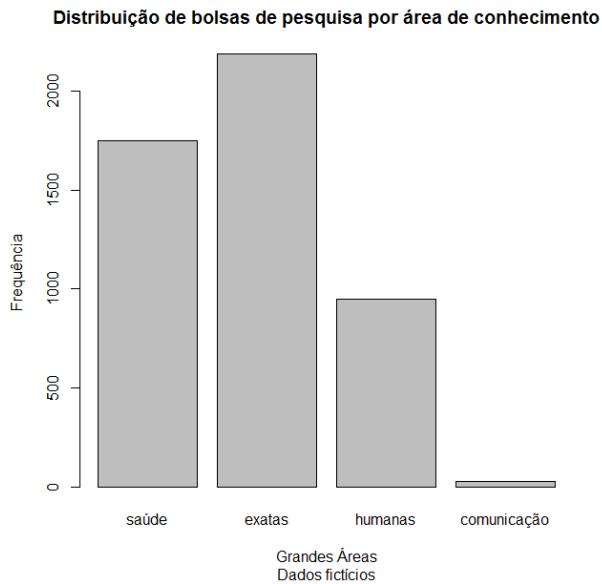


Figura 2-2: Gráfico de barras resultado do comando barplot com subtítulo

2.4.1 Modificando o tamanho da fonte

Usando o argumento `cex.nome`, podemos mudar o tamanho da fonte de diversas partes do gráfico. Se necessitar mudar o tamanho da fonte do título, dos eixos, do subtítulo ou dos números nos eixos acrescente o argumento

`cex.main= valor`, para o título

`cex.lab = valor`, para os eixos

`cex.axis = valor`, para os números dos eixos

`cex.sub = valor`, para o subtítulo que pode ser acrescentado abaixo do rótulo do eixo x

No R o valor padrão é 1, para aumentar use valor maior do que 1; para diminuir use valor menor do que 1. Confira o resultado da diminuição/aumento do título reproduzindo os comandos abaixo.

```
barplot(ya, names.arg=x, main="Distribuição de bolsas de pesquisa por área de conhecimento", cex.main=0.9)  
barplot(ya, names.arg=x, main="Distribuição de bolsas de pesquisa por área de conhecimento", cex.main=1.5)
```

Analogamente, para mudar o tamanho dos números do eixo y utilize o argumento *cex.axis*=valor.

```
barplot(ya, names.arg=x, main="Distribuição de bolsas de pesquisa por área de conhecimento", cex.axis=0.5)
```

Para mudar o tamanho da fonte dos nomes dos eixos, utilize o argumento *cex.lab*=valor.

```
barplot(ya, names.arg=x, main="Distribuição de bolsas de pesquisa por área de conhecimento", cex.lab=0.5)
```

2.4.2 Eliminando os eixos do gráfico

Podemos ainda retirar os eixos do gráfico, neste caso usa-se o argumento *axes=F*:

```
barplot(ya, names.arg=x, main="Distribuição de bolsas de pesquisa por área de conhecimento", axes=F)
```

Note que neste tipo de gráfico o eixo numérico corresponde apenas ao eixo y, portanto o argumento *axes=F* omitirá o eixo y.

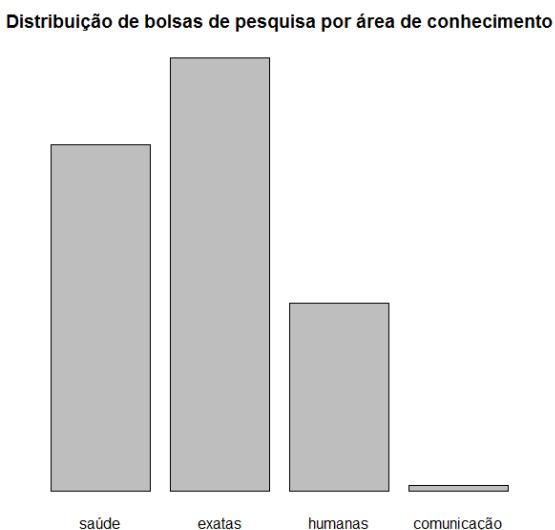


Figura 2-3: Gráfico de barras sem os eixos xy

2.4.3 Invertendo a posição das barras

Também podemos colocar as barras na posição horizontal, colocando o argumento `horiz=T`, neste caso os nomes dos eixos devem ser trocados, ou seja, `xlab` recebe o texto que anteriormente correspondia ao `ylab` e vice-versa.

```
x<-c("saúde", "exatas", "humanas", "comunicação")
ya<-c(1751, 2186, 947, 29)
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de
conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios",
horiz=T)
```

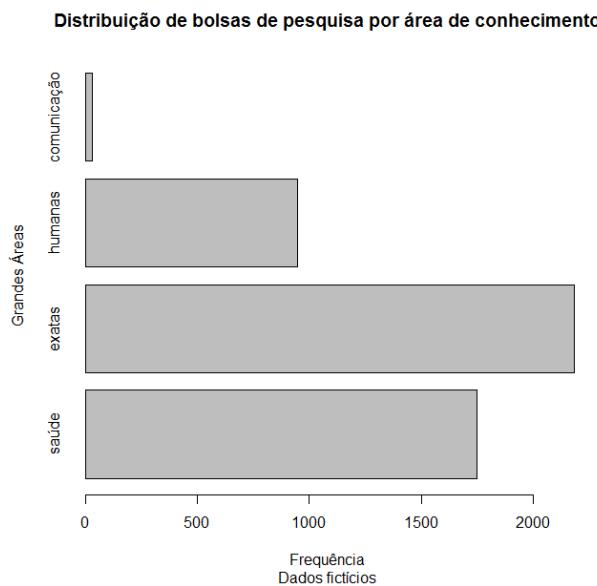


Figura 2-4: Gráfico de barras na posição horizontal

2.4.4 Cor e sombreamento das barras

Podemos preencher as barras com linhas de sombreamento ou com cor uniforme. Os gráficos acima foram preenchidos na cor cinza que é o padrão do R. Para escolher a linha de sombreamento das barras utilizamos `density`=valor, este valor representa a densidade de linhas por polegada e é um número positivo. É possível ainda definir o ângulo de inclinação

das linhas usando o argumento `angle=valor`, este valor variando de 0 a 360 no sentido anti-horário. Caso o ângulo não seja informado, o padrão será inclinação de 45°.

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", density=10)
```

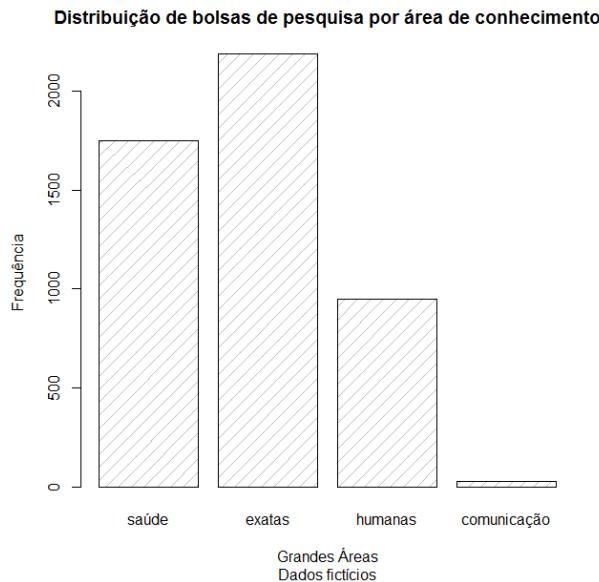


Figura 2-5: Gráfico de barras com preenchimento de linhas inclinadas

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", density=10, angle=90)
```

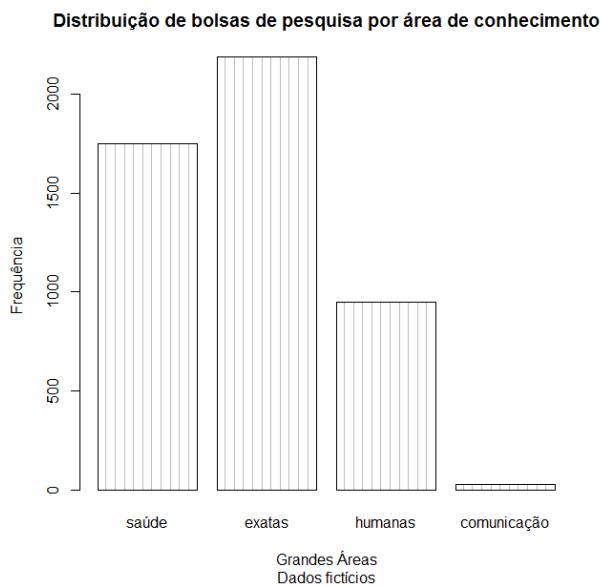


Figura 2-6: Gráfico de barras com preenchimento de linhas inclinadas à 90 graus.

Para preencher as barras com cores uniformes diferentes do padrão estabelecido, usamos o comando `col = "nome da cor"` ou `col = valor`, este valor representa o número da cor.

Para conhecer a paleta de cores utilize o comando `colors()` na linha de comando. Será exibido 657 nomes de cores possíveis. Por exemplo, `colors()[10]` corresponde à cor "aquamarine2".

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", col="aquamarine2")
```

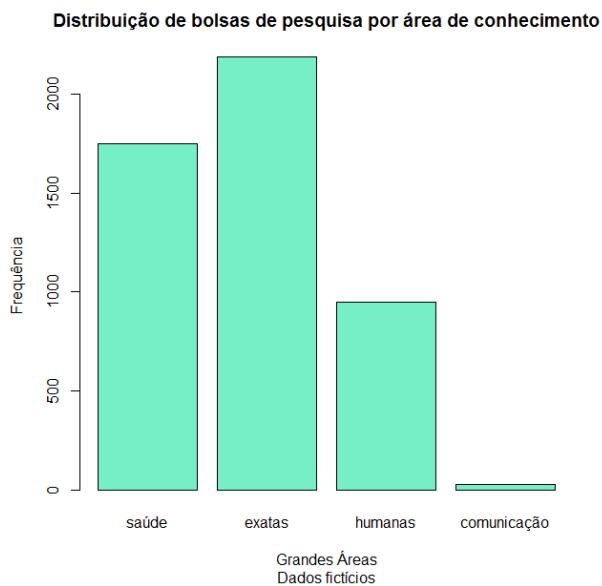


Figura 2-7: Gráfico de barras com preenchimento de cor através das opções de colors().

Podemos ainda colorir cada barra de uma cor, neste caso como temos 4 barras, utilizaremos uma sequência de cores que corresponderá à sequência das barras respectivamente. Podemos atribuir as cores pelos nomes ou por números de 1 a 8 que correspondem respectivamente às cores 1-preto, 2-vermelho, 3-verde, 4-azul, 5-ciano, 6-magenta, 7-amarelo e 8-cinza.

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", col=c(1, 2, 3, 4))
```

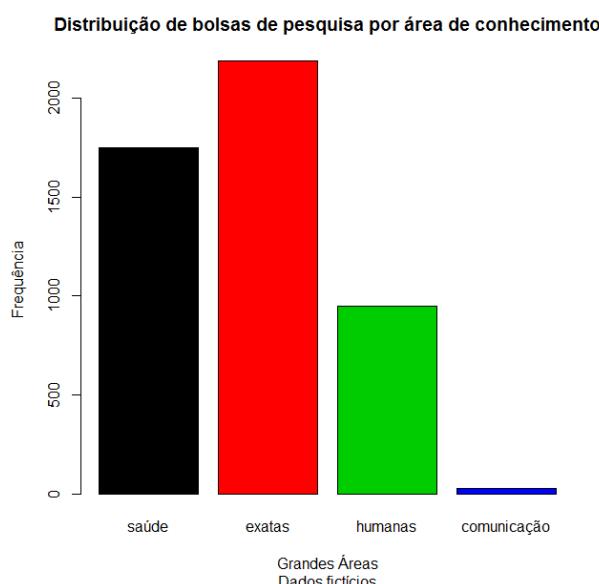
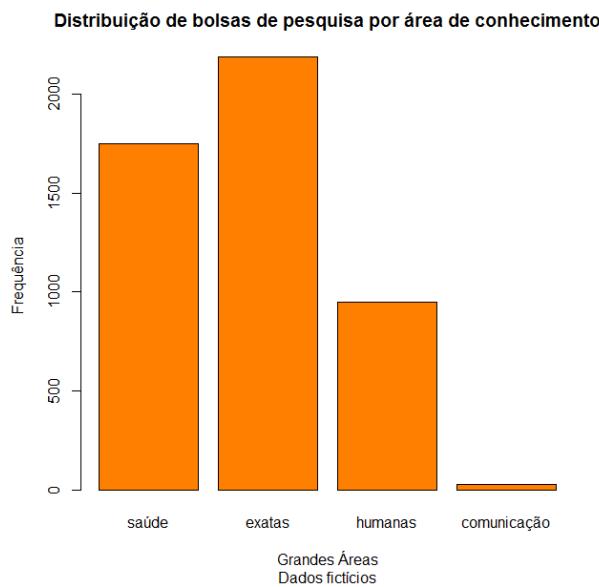


Figura 2-8: Gráfico de barras com preenchimento de cor através da atribuição de números.

É possível criar cores através do sistema RGB (Red,Green e Blue). Neste caso, valores entre 0 e 1 deverão ser atribuídos numa sequência tripla que corresponde ao Red, Green e Blue. Para se ter uma idéia de como o sistema funciona, a sequência 0, 0, 0 corresponde ao preto e a sequência 1, 1, 1 corresponde ao branco. Neste sistema é possível definir mais de 16 milhões de cores distintas.

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", col=rgb(1, 0.5, 0))
```



*Figura 2-9: Gráfico de barras com preenchimento de cor através do sistema *rgb*.*

2.4.5 Eliminando a linha de borda das barras

Podemos excluir a linha das bordas das barras acrescentando o argumento `border=NA`.

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", border=NA)
```

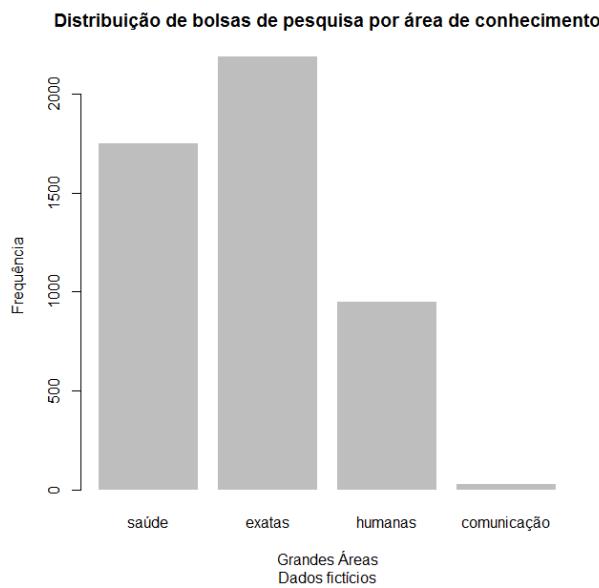


Figura 2-10: Gráfico de barras com preenchimento de cor e sem borda nas barras.

Para o caso de preenchimento com linhas, as bordas poderão ser omitidas:

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", density=10, border=NA)
```

Ou ainda poderão ser incluídas bordas no mesmo padrão das linhas de preenchimento das barras, basta acrescentar *border=T*.

```
barplot(ya, names.arg=x, main= "Distribuição de bolsas de pesquisa por área de conhecimento", xlab= "Grandes Áreas", ylab="Frequência", sub="Dados fictícios", density=10, col=5, border=T)
```

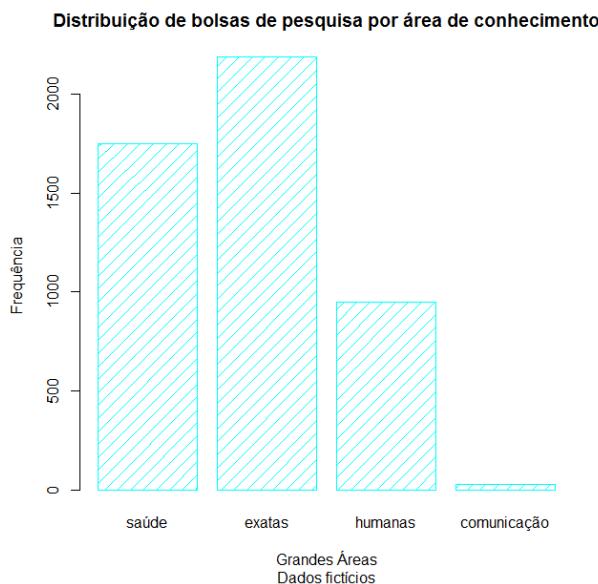


Figura 2-11: Gráfico de barras com preenchimento de linhas e borda nas barras no mesmo estilo.

2.4.6 Gráfico de Barras com duas variáveis

Também podemos criar o gráfico de barras de duas variáveis, um ao lado do outro, na mesma janela gráfica. Para isso, cada variável deverá ser armazenada em um vetor que chamaremos de ya e yb. Será necessário criarmos uma matriz com duas colunas, uma para cada variável e 4 linhas, uma para cada grande área do nosso exemplo. Chamaremos tal matriz de Y e nomearemos as linhas e colunas desta matriz conforme *script* abaixo:

Script Gráfico de Barras para duas variáveis

```
x<-c("saúde","exatas","humanas","comunicação")
```

```

ya<-c(1751, 2186, 947, 29)
yb<-c(2528, 2132, 1843, 280)
Y<-matrix(c(ya,yb), nrow=4, ncol=2, dimnames=list(x, c("Intituição A", "Instituição
B")))
barplot(Y, main= "Distribuição de bolsas de pesquisa por área de conhecimento",
xlab= "Grandes Áreas", ylab="Frequência")

```

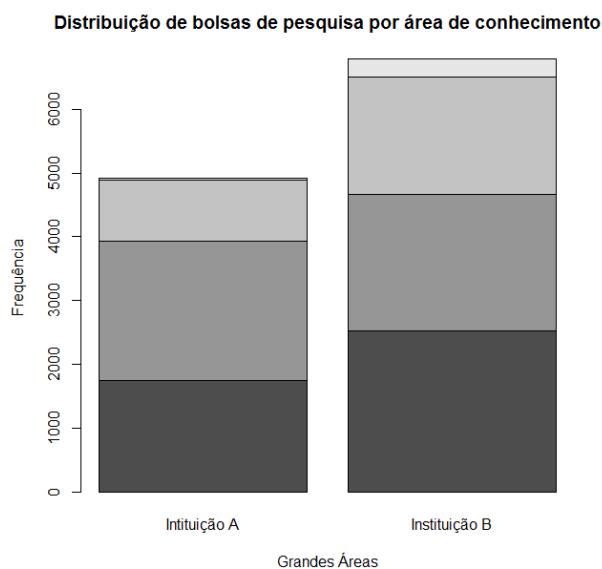


Figura 2-12: Gráfico de barras com Duas variáveis: Instituição A e B, barras empilhadas.

O padrão do comando *barplot* aplicado a uma matriz é empilhar todas as categorias, acumulando as frequências. Às vezes, essa maneira não é a mais adequada para visualizar o que se deseja, então poderemos usar o argumento *biside=T* para obtermos barras lado a lado:

Script Gráfico de Barras para duas variáveis sem empilhamento

```

x<-c("saúde", "exatas", "humanas", "comunicação")
ya<-c(1751, 2186, 947, 29)
yb<-c(2528, 2132, 1843, 280)
Y<-matrix(c(ya,yb),nrow=4,ncol=2, dimnames=list(x, c("Intituição A", "Instituição
B")))
barplot(Y, main= "Distribuição de bolsas de pesquisa por área de conhecimento",
xlab= "Grandes Áreas", ylab="Frequência", beside=T)

```

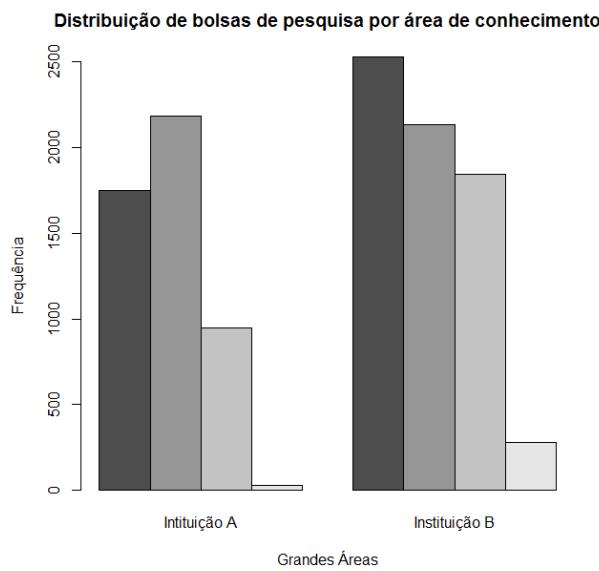


Figura 2-13: Gráfico de barras com Duas variáveis: Instituição A e B, barras lado a lado

Para finalizarmos a apresentação do gráfico, adicionaremos legenda e subtítulo citando a fonte.

Script Gráfico de Barras para duas variáveis com legenda

```
x<-c("saúde","exatas","humanas","comunicação")
ya<-c(1751, 2186, 947, 29)
yb<-c(2528, 2132, 1843, 280)
Y<-matrix(c(ya, yb), nrow=4, ncol=2, dimnames=list(x, c("Intituição A", "Instituição B")))
barplot(Y, main= "Distribuição de bolsas de pesquisa por área de conhecimento",
xlab= "Grandes Áreas", ylab="Frequência", beside=T, legend.text=rownames(Y),
sub="Fonte: Dados fictícios")
```

Pode ocorrer de a legenda sobrepor as barras do gráfico, neste caso sugerimos o artifício de aumentar o eixo y utilizando o argumento `ylim=c(valor inicial, valor final)`.

```
barplot(Y, main= "Distribuição de bolsas de pesquisa por área de conhecimento",
xlab= "Grandes Áreas", ylab="Frequência", beside=T, legend.text=rownames(Y),
sub="Fonte: Dados fictícios", ylim=c(0, 3000))
```

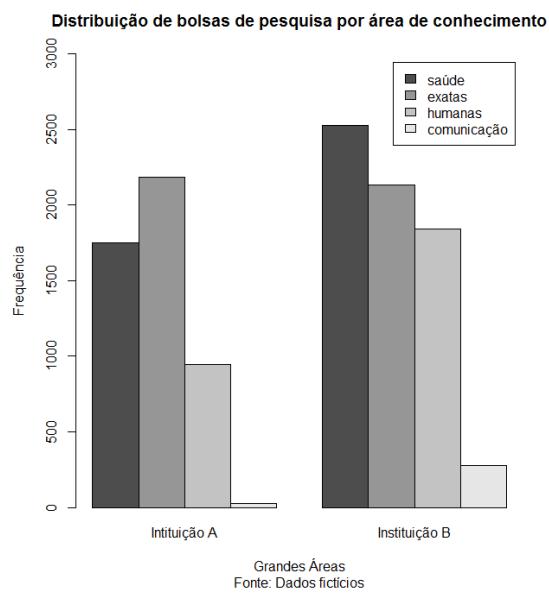


Figura 2-14: Gráfico de barras com Duas variáveis: Intituição A e B, barras lado a lado com legenda

2.4.7 Hora de Praticar

1. Com base nos dados da tabela 2.1 construa um gráfico de barras que mostra a relação de alunos/professor em cada tipo de escola. Neste caso será necessário obter os valores da nova variável que é o resultado da coluna 3 dividido pela coluna 2. Coloque título adequado, legenda, e nomes nos eixos.
2. Com base na questão 1, refaça o gráfico de barras com a barras no sentido horizontal.
3. Refaça o gráfico da questão 1 utilizando sombreamento de densidade 60.
4. Refaça o gráfico da questão 1 utilizando sombreamento de densidade 50 e ângulo das linhas de sombreamento de 75 graus.
5. 2- O vetor peso indica o peso médio de pintinhos com 2,4,6,8 e 10 dias de nascido respectivamente.
peso=(42,51,59,64,76).

Considere cada tempo de nascido como sendo uma categoria. Estabeleça um gráfico de barras para o peso médio dos pintinhos em cada categoria. Coloque título, legenda e nomes nos eixos.

6. Refaça o gráfico da questão 5 utilizando o seguinte esquema de cores: “blue”, “pink”, “yellow”, ”green”, ”red”.
7. Refaça o gráfico da questão 6 utilizando barras horizontais.
8. Numa central telefônica de uma grande empresa, havia a sensação de saturação do sistema utilizado. Para melhor representar o que ocorria foi realizado um acompanhamento com as telefonistas que teriam que responder aos problemas em que números ocorriam e lançá-los na *Lista de Verificação*. Tab3 resume os dados desta lista. Apresente um gráfico de barras para estes dados com barras verticais com sombreamento de densidade 80 e ângulo das linhas de 120 graus.

Tipo de defeito	Número de ocorrência
Linha Ruidosa	250
Linha Aberta	110
Alarme	85
Não Responde	45
Não Toca	25

9. Considere os dados da Tabela sobre a freqüência para um grupo de fumantes de cigarro e um grupo de não-fumantes em diversos níveis séricos de cotinina. Construa um gráfico de barras semelhante ao da figura 2.9

<i>Nível de Cotinina (ng/ml)</i>	<i>Fumantes</i>	<i>Não-fumantes</i>
0 – 13	78	3300
14 – 49	133	72
50 – 99	142	23
100 – 149	206	15
150 – 199	197	7
200 – 249	220	8
250 – 299	151	9
300 +	412	11

2.5 Gráfico de setor

Os gráficos de setor (ou pizza) são representados por círculos divididos proporcionalmente de acordo com os dados do fenômeno ou do processo a ser representado. Os valores são expressos em números ou em percentuais (%). São muito úteis para termos uma visão do tamanho da fatia correspondente a uma categoria quando comparada com o todo. Não é adequado quando se quer utilizá-lo para representar muitas categorias ao mesmo tempo.

Os dados da tabela 2.2 são referentes a frota de veículos no município de Niterói no ano de 2009. Utilizaremos estes dados para fazermos o gráfico.

Tabela 2.2: Frota de principais tipos de veículo do município de Niterói em 2009.

Tipo de veículo	Frota
Automóveis	151882
Motocicletas	19627
Caminhonetas	8058
Motonetas	3201
Ônibus	1969
Caminhões	1895
Outros	1268

Fonte: IBGE, Cidades - 2009

O primeiro passo na construção do gráfico é ter os dados armazenados em objeto apropriado nomeando cada posição do vetor. Os nomes de cada posição aparecerão no gráfico conforme pode ser visto na figura 2.15.

Script Básico para gráfico de setores

```
frota<-c(151882, 19627, 8058, 1895, 3201, 1969, 1268)  
names(frota)=c("automóveis", "motocicletas", "caminhonetas", "motonetas",  
"ônibus", "caminhões", "outros")  
pie(frota)
```

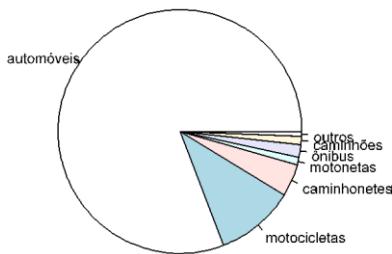


Figura 2-15: Gráfico de setores para os dados da frota de veículos de Niterói em 2009 sem acabamento.

O gráfico da figura 2.15 não é informativo, pois o leitor não sabe do que se trata. Para que se torne compreensível é necessário realizar diversas melhorias como colocação de título, e os percentuais que representa cada setor.

O título pode ser adicionado após a janela gráfica ser exibida. Neste caso utilizamos o comando `title("texto")` na sequência de comandos do script básico. Note aqui que o gráfico já está construído, apenas acrescentamos o título.

```
title("Frota 2009 - Niterói_RJ")
```

Para colocar as porcentagens utilizamos o argumento `labels`. Observe que nesta sequência o título é colocado como argumento da função `pie`. Aqui é necessário que se coloque uma legenda para que o leitor comprehenda o gráfico.

Script para gráfico de setores com título e legenda

```
frota<-c(151882, 19627, 8058, 1895, 3201, 1969, 1268)
names(frota)=c("automóveis", "motocicletas", "caminhonetes", "motonetas",
"ônibus", "caminhões", "outros")
porc<-round(frota*100/sum(frota), 2) #arredonda a porcentagem para 2 dígitos
significativos)
rotulos<-paste((",",porc,"%"), sep="")
pie(frota, main="Frota 2009 - Niterói_RJ", labels=rotulos, col=rainbow(7))
legend(1, 1, names(frota), col = rainbow(7), pch=rep(20,6))
```

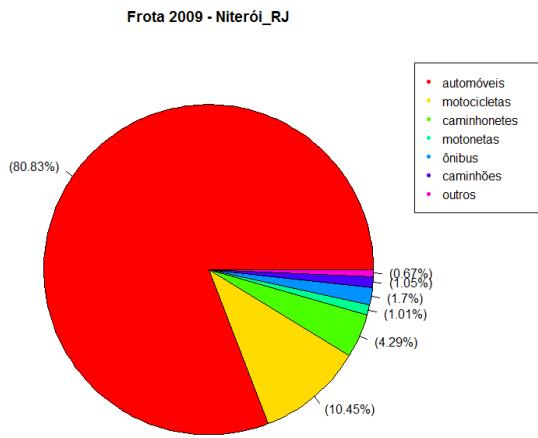


Figura 2-16: Gráfico de setores para os dados da frota de veículos de Niterói em 2009 com título e legenda.

Se desejar o nome e a porcentagem, utilize a sequência de comandos a seguir, diminuindo o tamanho da fonte e alterando o esquema de cores para melhor adequação:

```
rotulos<-paste(names(frota),"(","porc,"%)", sep="")
pie(frota, main="Frota 2009 - Niterói_RJ", labels=rotulos, cex=0.7, col=rainbow(7))
```

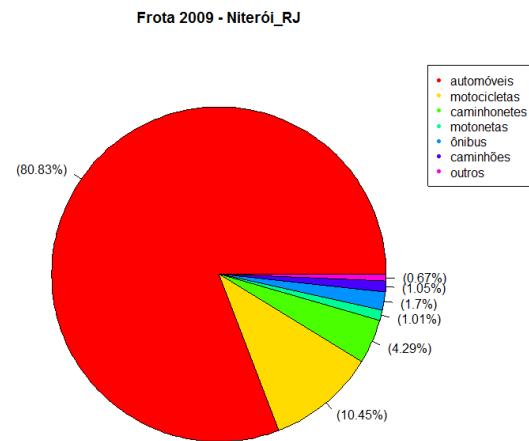


Figura 2-17: Gráfico de setores para os dados da frota de veículos de Niterói em 2009 com título e rótulos de porcentagem

Podemos colocar os nomes manualmente, mas só é bom utilizarmos este recurso no caso de algum nome ficar sobreposto a outro, pois dá um pouco de trabalho, dessa forma:

```
pie(frota, main="Frota 2009 - Niterói_RJ", labels=NA)
text(locator (length(names(frota))), rotulos)
```

Após este último comando é necessário que se clique na janela do gráfico onde se deseja que os rótulos sejam impressos. O gráfico só ficará pronto depois de clicar 7 vezes, pois este é o número de setores do gráfico. Pode-se repetir este procedimento até obter o gráfico desejado.

Alguns ajustes podem ser necessários. Para mudar o tamanho da fonte do título use

```
cex.main=valor
```

```
pie(frota, main="Frota 2009 - Niterói_RJ", cex.main=1.2)
```

Para mudar o tamanho da fonte dos nomes das variáveis use *cex=valor*.

```
pie(frota, cex=0.9)
```

Para girar o gráfico, use *init.angle=valor*.

```
pie(frota, main="Frota 2009 - Niterói_RJ", init.angle=180)
```

2.5.1 Hora de Praticar

1. Numa central telefônica de uma grande empresa. Havia a sensação de saturação do sistema utilizado. Para melhor representar o que ocorria foi realizado um acompanhamento com as telefonistas que teriam que responder aos problemas em que números ocorriam e lançá-los na *Lista de Verificação*. Tab3 resume os dados desta lista. Apresente um gráfico de setores para estes dados semelhante ao gráfico da figura 2.12.

Tipo de Defeito	Número de Ocorrência
Linha Ruidosa	250
Linha Aberta	110
Alarme	85
Não Responde	45
Não Toca	25

2. Refaça o gráfico anterior semelhante ao apresentado na figura 2.13
3. Os dados a seguir nos informam as despesas pessoais dos Estados Unidos (em bilhões de dólares), com Comida e cigarro, domésticos, remédios e saúde, cuidados pessoais

e educação privada, respectivamente. Faça o gráfico de setor semelhante ao da figura 2.12.

despesas=c(22200,10500,3530,1040,341)

4. Refaça o gráfico anterior semelhante ao apresentado na figura 2.13.

2.6 Histograma

Um histograma é uma representação gráfica da distribuição de frequências de uma massa de medições, normalmente um gráfico de barras verticais. Tal gráfico é composto por retângulos justapostos em que a base de cada um deles corresponde ao intervalo de classe e a sua altura à respectiva frequência.

O primeiro passo na construção é ter os dados armazenados em objeto apropriado. Por praticidade utilizaremos dados disponíveis no R o *data(mtcars)*.

```
data(mtcars)
```

```
mtcars[1:6,] #visualizando parte dos dados até a sexta linha
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Vamos explorar os dados da variável *mpg* (primeira coluna) que são os dados do consumo de milhas por galão de diversos tipos de carros.

Para fazermos o histograma usamos o comando *hist()*.

```
attach(mtcars)
```

```
hist(mpg)
```

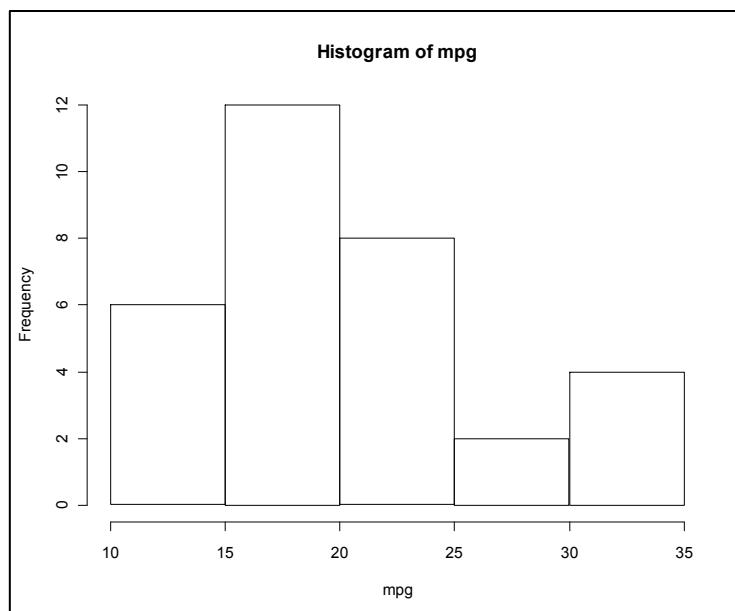


Figura 2-18: Histograma do consumo de milhas por galão

Podemos colocar nosso gráfico em frequência relativa:

```
hist(mpg, freq=FALSE)
```

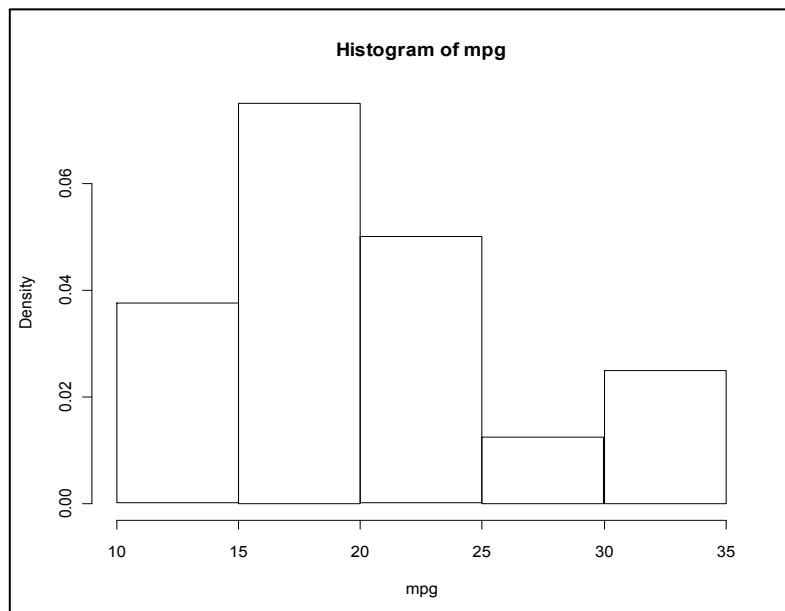


Figura 2-19: Histograma da frequência relativa do consumo de milhas por galão
Colocar linhas de sombreamento:

```
hist(mpg, density=30)
```

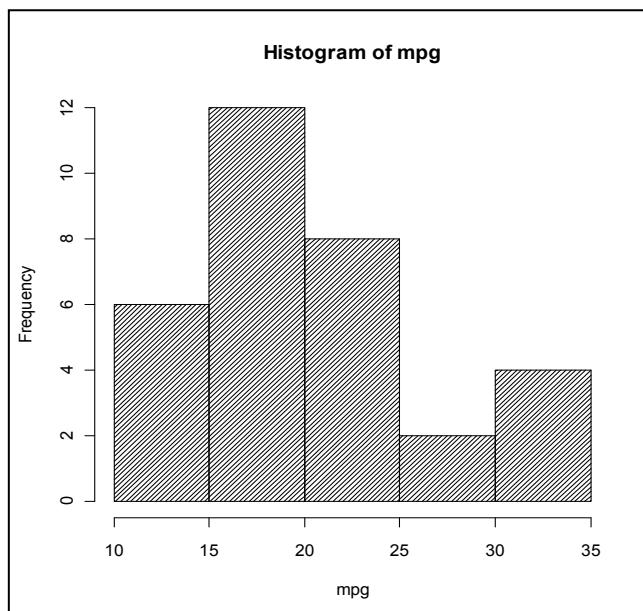


Figura 2-20: Histograma com linhas de sombreamento

Podemos mudar a angulação dessa linha:

```
hist(mpg, density=1, angle=80)
```

Para colocar barras com cores:

```
hist(mpg, col=c("blue","red","orange","green","pink"))
```

Podemos alterar as cores das bordas:

```
hist(mpg, border=c("blue","red","orange","green","pink"))
```

Ou retirar as bordas:

```
hist(mpg, col=c("blue","red","orange","green","pink"), border=FALSE)
```

O histograma, já vem com um título e nomes nos eixos, mas nós também podemos modificá-los:

```
hist(mpg, main= "Histograma de milhas por galão", xlab="MPG",  
ylab="frequência")
```

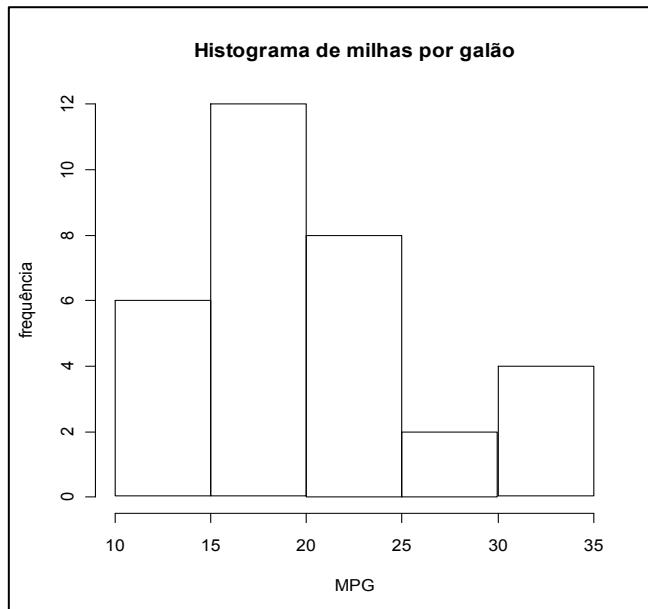


Figura 2-21: Histograma com título principal e rótulos nos eixos.

Podemos definir a quantidade de classes do gráfico:

```
hist(mpg, breaks=3)
```

E também fazer o ramo de folhas:

```
stem(mpg)
```

The decimal point is at the |

10 | 44

12 | 3

14 | 3702258

16 | 438

18 | 17227

20 | 00445

22 | 88

24 | 4

26 | 03

28 |

30 | 44

32 | 49

2.6.1 Hora de Praticar

1. Os dados são referentes às temperaturas diárias dos meses de maio e setembro, respectivamente, em Fahrenheit, na cidade de Nova York em 1973.

```
tempm=c(67,72,74,62,56,66,65,59,61,69,74,69,66,68,58,64,66,57,68,62,59,73,61,61,57,58  
,57,67,81,79,76)
```

```
temps=c(91,92,93,93,87,84,80,78,75,73,81,76,77,71,71,78,67,76,68,82,64,71,81,69,63,70,  
77,75,76,68)
```

- a. Faça o histograma das temperaturas do mês de maio. Coloque título e linhas de sombreamento de densidade 30.
- b. Faça o histograma das temperaturas do mês de setembro. Coloque título e cor = “Violet”.
- c. Converta as temperaturas do mês de maio para graus Celsius através da expressão $^{\circ}\text{C} = (^{\circ}\text{F} - 32) / 1,8$. Faça o histograma da frequência relativa, coloque título, sombreamento de densidade 25 e cor = “dark blue”

2.7 Boxplot

O *boxplot* é um gráfico que possibilita representar a distribuição de um conjunto de dados com base em alguns de seus parâmetros descritivos, quais sejam: a mediana (q_2), o quartil inferior (q_1), o quartil superior (q_3) e do intervalo interquartil ($IQR = q_3 - q_1$). .

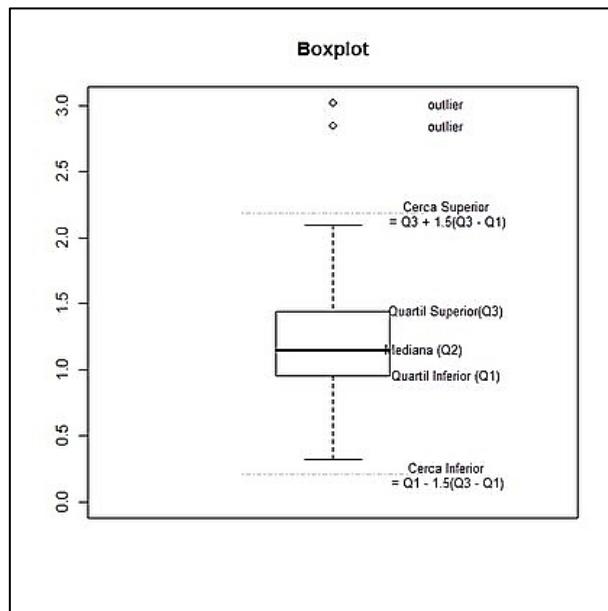


Figura 2-22: Box Plot ilustrativo.

Vamos usar os dados do próprio R, sobre uma amostra de 48 pedras de uma reserva de petróleo, onde temos a área, perímetro, formato e permeabilidade. Para exemplificarmos, usaremos a variável formato (*shape*) que é dada pelo perímetro dividido pela raiz quadrada da área da pedra.

Script para BoxPlot simples

```
data(rock)  
attach(rock)  
boxplot(shape)
```

Adicionando título ao gráfico:

```
boxplot(shape, main="BoxPlot do formato da pedra" , ylab="perimetro/sqrt(area)")
```

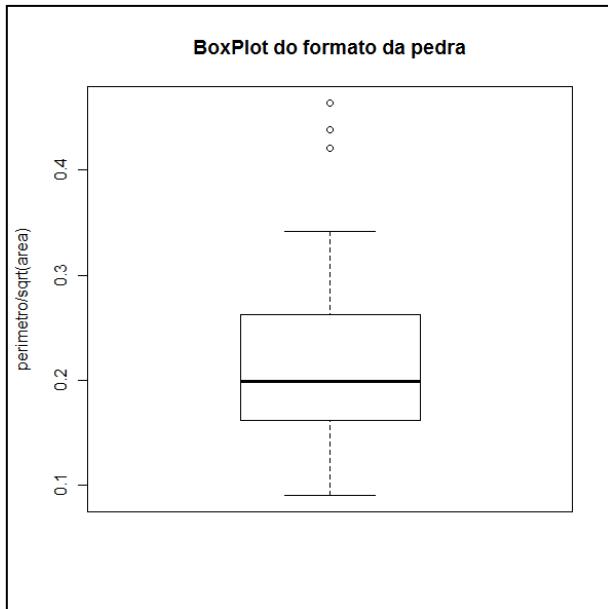


Figura 2-23: Box Plot com título

Se quisermos que as cercas sejam os valores máximo e mínimo utilizamos range=0:

```
boxplot(shape, range=0)
```

Se quisermos que o *outlier* não apareça no gráfico, utilizamos o seguinte comando:

```
boxplot(shape, outline=FALSE)
```

Podemos fazer o *boxplot* para vários grupos de observação, facilitando a comparação entre os grupos. Vamos utilizar os dados de *data(PlantGrowth)*

Estes dados são resultados de um experimento para comparar rendimentos (medido pelo peso seco de plantas) obtidos em um controle e duas condições de tratamento diferentes.

```
data(PlantGrowth)
```

```
attach(PlantGrowth)
```

```
boxplot(weight~group)
```

```
title("Boxplot para rendimento de plantas segundo o tratamento", xlab = "tratamento", ylab = "peso")
```

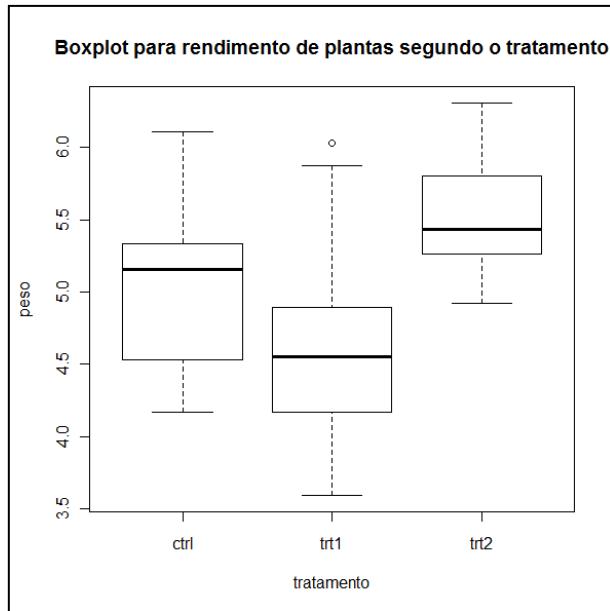


Figura 2-24: Box Plot para vários grupos

Aprimoramentos:

Se `varwidth=TRUE`, as caixas são desenhadas com larguras proporcionais à raiz quadrada do número de observações. Este caso se aplica a grupos de tamanhos diferentes.

```
x1<-c(2, 3, 4, 4, 5, 6, 8, 9, 7, 7, 7, 7, 6)
```

```
x2<-c(1, 1, 1, 1, 2, 3, 4, 3, 2, 3, 4, 5, 6, 3, 4, 3, 2, 2, 2, 3, 4, 5, 6, 3, 2, 1, 1, 1, 2, 3, 4)
```

```
boxplot(x1, x2, varwidth=TRUE)
```

Se desejamos nomear os grupos, utilizamos o argumento *names*:

```
boxplot(x1, x2, names=c("grupo1","grupo2"))
```

Para modificar a largura da caixa para mais estreita utilizando o argumento *boxwex*:

```
boxplot(x1, x2, boxwex=0.3)
```

Para ajeitar o tamanho das linhas limites:

```
boxplot(x1, x2, staplewex=0.1)
```

Para colocar cor nas bordas:

```
boxplot(shape, border="red")
```

Para preencher a cor de dentro da caixa:

```
boxplot(shape, col="blue")
```

Podemos colocar o gráfico no sentido horizontal:

```
boxplot(shape, horizontal= TRUE, main= "BoxPlot do formato da pedra", xlab= "valor")
```

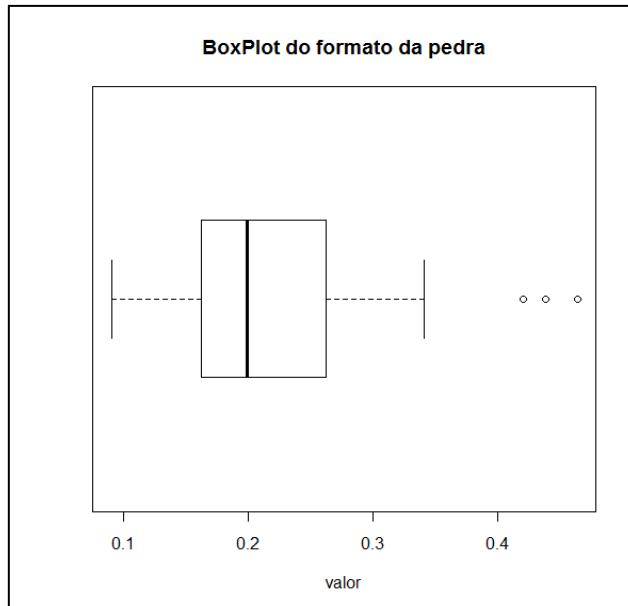


Figura 2-25: Box Plot horizontal

Também podemos verificar todos os cálculos utilizados para a formação do gráfico, seus limites, *outliers*, etc:

```
boxplot.stats(shape)
```

Resultado de saída do comando:

```
$stats
```

```
[1] 0.0903296 0.1621295 0.1988620 0.2626890 0.3412730
```

```
$n
```

```
[1] 48
```

```
$conf
```

```
[1] 0.1759291 0.2217949
```

```
$out
```

```
[1] 0.438712 0.464125 0.420477
```

2.7.1 Hora de Praticar

1. Utilize os dados de contagens de insetos em unidades experimentais agrícolas tratados com inseticidas diferentes disponível no R em InsectSprays e construa um boxplot para os 6 tipos de inseticidas. Coloque título adequado, e caixas preenchidas com a cor lightgray.
2. Refaça o gráfico anterior sem a presença dos *outliers*.
3. Refaça o primeiro gráfico reduzindo a largura das caixas.
4. Utilize os dados disponíveis no R em *rock* (veja detalhes sobre estes dados na pag 21) e defina dois vetores:

x é o vetor de perímetros cujas pedras possuam área menor ou igual à área média, ou seja, `x<-peri[area<=mean(area)]`

y é o vetor de perímetros cujas pedras possuam área maior à área média, ou seja, `y<-peri[área>mean(area)]`

Faça o boxplot de x e y semelhante ao da figura 2.22

5. Refaça o gráfico anterior utilizando o argumento `varwidth=TRUE`.

2.8 Gráfico de dispersão

Os diagramas de dispersão são representações de duas variáveis que são organizadas em um gráfico, para observar o padrão de relacionamento entre as mesmas. É um método gráfico que permite verificar a existência ou não de relação entre duas variáveis de natureza quantitativa.

Os dados que iremos analisar são medidas da circunferência em mm do tronco de pés de laranjeiras e sua respectiva idade em dias. Com esses dados, criaremos o gráfico de dispersão para observar se há relação entre o tamanho da circunferência com a idade da árvore.

Script para gráfico de dispersão

```
data(Orange)
attach(Orange)
#Vamos considerar os 7 primeiros valores que correspondem a uma única árvore:
plot(age[1:7], circumference[1:7], xlab="idade em dias", ylab="circunferência em mm", main="Dispersão entre idade e circunferência")
```

É interessante colocarmos a reta de regressão linear no mesmo gráfico. Para isso, acrescentamos mais uma linha de comando no *script* acima:

Script para gráfico de dispersão com reta de regressão

```
plot(age[1:7], circumference[1:7], xlab="idade em dias", ylab="circunferência em mm", main="Dispersão entre idade e circunferência do tronco de pés de laranjeiras")
abline(lm(circumference[1:7] ~ age[1:7]), col=2) #acrescenta a reta de regressão na cor vermelha (col=2).
text(450, 100, "reta de regressão") # acrescenta o texto entre aspas na posição x=450 e y=100.
```

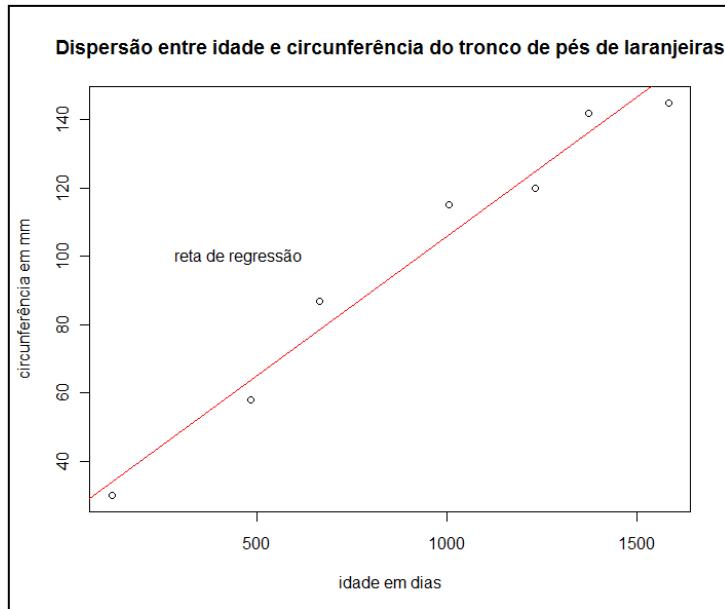


Figura 2-26: Diagrama de Dispersão e a reta de regressão

A largura da linha que contorna os pontos do gráfico pode ser mudada com o argumento *lwd*,

```
plot(age[1:7], circumference[1:7], lwd=4) #Teste vários números para ver a diferença.
```

Podemos mudar as bolas por outros símbolos. Para cada número temos uma forma diferente:

```
plot(age[1:7], circumference[1:7], pch=20). #Teste vários números para ver a diferença.
```

Podemos colocar vários símbolos num mesmo gráfico:

```
plot(age[1:7], circumference[1:7], pch=1:7)
```

Podemos criar alguns símbolos, por exemplo:

```
plot(age[1:7], circumference[1:7], pch="@")
```

```
plot(age[1:7], circumference[1:7], pch="&")
```

Colocando cores:

```
plot(age[1:7], circumference[1:7], lwd=4, col="orange")
```

Colocando subtítulo:

```
plot(age[1:7], circumference[1:7], sub="Desde 1968")
```

E criar um texto também na posição escolhida (400, 80):

```
text(400, 80, "Isso aqui pode ser escrito aqui")
```

Lembrando que para colocar
título principal : *main* = “texto”
título nos eixos: *xlab*=”texto” ou *ylab*=”texto”
subtítulo: *sub*=”texto”

Podemos ainda mudar os limites dos eixos:

```
plot(age[1:7], circumference[1:7], xlim=range(100:1600), ylim=range(20:180))
```

Ainda podemos colocar mais de um gráfico numa mesma janela:

Script para 6 gráficos de dispersão em uma única janela

```
par(mfrow=c(2, 3)) #assim podemos fazer 6 gráficos em 2 linhas e 3 colunas.  
  
plot(age[Tree==1], circumference[Tree==1], xlab= "idade", ylab= "circunferência",  
sub= "árvore1")  
  
plot(age[Tree==2], circumference[Tree==2], xlab= "idade", ylab= "circunferência",  
sub= "árvore2")  
  
plot(age[Tree==3], circumference[Tree==3], xlab= "idade", ylab= "circunferência",  
sub= "árvore3")  
  
plot(age[Tree==4], circumference[Tree==4], xlab= "idade", ylab= "circunferência",  
sub= "árvore4")  
  
plot(age[Tree==5], circumference[Tree==5], xlab= "idade", ylab= "circunferência",  
sub="árvore5")  
  
plot(age, circumference, xlab="idade", ylab="circunferência", sub="árvores")
```

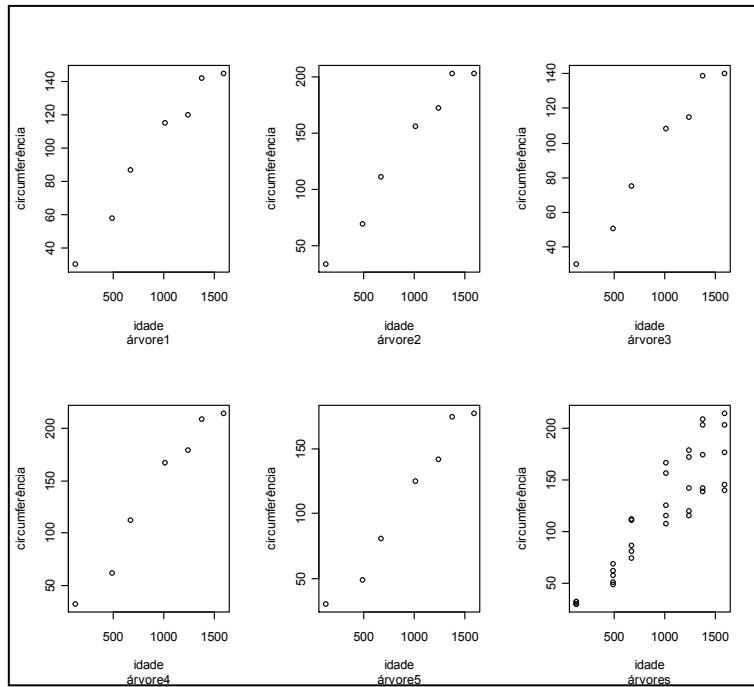


Figura 2-27: Diagrama de Dispersão vários lado a lado

2.8.1 Hora de Praticar

1- Um biólogo foi ao campo e contou o número de sapos em 20 locais. Ele também anotou a umidade e a temperatura em cada local. Faça dois gráficos de dispersão para mostrar a relação do número de sapos com as variáveis temperatura e umidade. Use a função *par()* para dividir a janela em duas.

Os dados são:

sapos 6-5-10-11-26-16-17-37-18-21-22-15-24-25-29-31-32-13-39-40

umid 62-24-21-30-34-36-41-48-56-74-57-46-58-61-68-76-79-33-85-86

temp 31-23-28-30-15-16-24-27-18-10-17-13-25-22-34-12-29-35-26-19

2- Um biólogo interessado em saber se o número de aves está relacionado ao número de uma determinada espécie de árvore, realizou amostras em 10 locais. Os valores obtidos foram:

aves<-c(22,28,37,34,13,24,39,5,33,32)

árvores<-c(25,26,40,30,10,20,35,8,35,28)

Faça um gráfico que mostra a relação entre o número de aves e o número de árvores. Um colega coletou mais dados sobre aves e árvores, em outra área, que podemos aproveitar. Os dados são:

```
arvores2<-c(6,17,18,11,6,15,20,16,12,15)
```

```
aves2<-c(7,15,12,14,4,14,16,60,13,16)
```

Inclua estes novos pontos no gráfico com um símbolo diferente e cor azul.

3 – Repita o gráfico do exercício 2 acima e faça as seguintes modificações.

Coloque um título no gráfico

Use Bolinhas cheias e azuis como símbolo.

Coloque as legendas no eixo-x e no eixo-y.

O padrão do R é fazer uma —caixa entorno do gráfico, faça uma alteração para que apareça apenas as linhas do eixo-x e do eixo-y. Veja *bty* em *?par*

4-Foram levantados os tempos (em horas por semana) que 20 alunos de uma escola da rede pública gastam na Internet e praticando atividades físicas. Os resultados deste levantamento são apresentados a seguir:

Aluno 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Internet 30 8 20 15 12 4 40 25 20 14 22 2 15 18 30 20 24 15 8 8

At. Físicas 2 10 5 5 8 15 0 4 5 10 2 15 6 4 2 6 4 10 12 15

Construa um gráfico de linhas para esse par de variáveis. Com base neste gráfico, você acredita que há relação entre as horas dedicadas às duas atividades?

2.9 Gráfico de linhas

Este gráfico é útil para plotar uma série de dados ao longo do tempo ou para ligar pontos por linhas.

Vamos ilustrar uma situação onde desejamos plotar o gráfico de duas funções, x ao cubo e $-x$ ao cubo no mesmo gráfico. Neste caso, definiremos inicialmente uma janela gráfica usando o argumento `type="n"`. Com este argumento um "gráfico em branco" é criado, são ajustadas apenas as margens e eixos do gráfico e o restante é deixado em branco. A seguir adicionam-se linhas e pontos desejados. Você deve fornecer coordenadas x e y que cubram a amplitude de valores de todos os elementos que você deseja adicionar ao gráfico.

Script para gráfico de linhas, duas funções no mesmo gráfico

```
x <- 0:20  
y <- x**3  
plot(c(0,20), c(-8000, 8000), type='n', xlab=NA, ylab=NA)  
lines(x, y)  
lines(x, -y, col='red')  
title("Gráfico de duas funções", xlab="valores de x", ylab="valores de y")
```

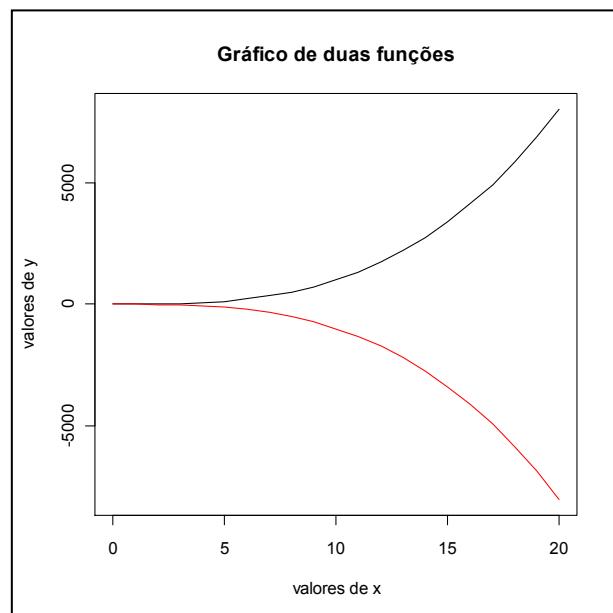


Figura 2-28: Gráfico de duas funções

Vamos mostrar agora o gráfico de uma série de valores ao longo do tempo.

Considere os dados da tabela 2.3, correspondentes à temperatura média obtida em Ithaca (NY):

Tabela 2.3: Temperatura Média Mensal de Ithaca - NY.

Mês	Temperatura °C
1	-5.44
2	-5.17
3	0.11
4	6.89
5	12.67
6	17.94
7	20.44
8	19.5
9	15.67
10	9.72
11	4.06
12	-2.56

Script para gráfico de linha

```
mes<-1:12
temperatura<-c(-5.44, -5.17, 0.11, 6.89, 12.67, 17.94, 20.44, 19.5, 15.67, 9.72, 4.06, -2.56)
plot(mes, temperatura, type='l')
title("Gráfico da temperatura ao longo dos meses")
```

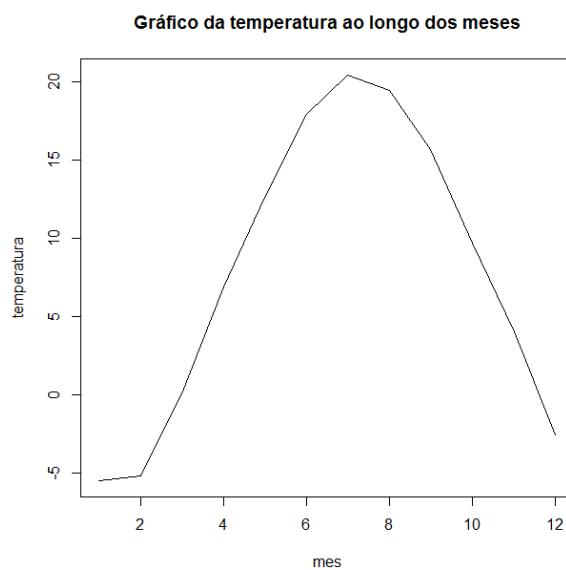


Figura 2-29: Gráfico de linha da série de temperatura de Ithaca - NY

Ou se quiser marcar os pontos substitua o comando acima por:

```
plot(mes, temperatura, type='b')
title("Gráfico da temperatura ao longo dos meses")
```

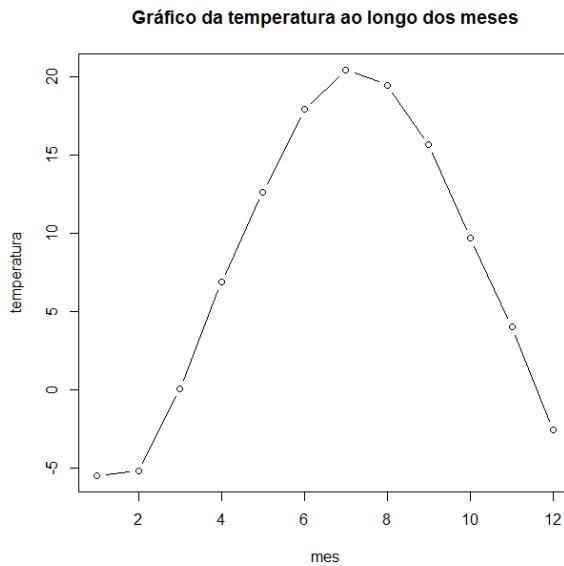


Figura 2-30: Gráfico de linha e pontos da série de temperatura de Ithaca - NY

Podemos utilizar comandos já vistos anteriormente para adicionar título aos eixos, mudar tamanho da fonte, mudar cor, símbolos, adicionar subtítulo, mudar os limites dos eixos, etc.

main=“texto” adiciona título ao gráfico

xlab ou *ylab* = “texto” adiciona título ao eixo x ou y

cex = número altera o tamanho da fonte

type = "p" para pontos,

 "l" para linhas,

 "b" para pontos e linhas,

 "c" para linhas descontínuas nos pontos,

 "o" para pontos sobre as linhas,

 "n" para nenhum gráfico, apenas a janela.

col = “nome” ou número da cor

pch = número (0 a 25)

lwd= número (controla a espessura da linha)

Podemos ainda grafar duas ou mais séries no mesmo gráfico:

Script para plotar duas séries temporais em gráfico de linha

```
ano<-2001:2009
```

```
tri1<-c(72.8, 66.2, 69.2, 65.9, 62.4, 67.8, 61.3, 68.5, 70.4)
```

```
tri2<-c(60.6, 53.7, 55.3, 56.7, 56.4, 57.8, 57.5, 59.8, 63.3)
```

```
plot(ano, tri1, type="l", main="Taxa de ocupação por trimestre dos hotéis -  
Município do Rio de Janeiro", xlab="ano", ylab="Taxa de ocupação %", col="blue",  
ylim=c(50,80))
```

```
lines(ano, tri2, col="red")
```

É sempre possível mudar o tipo de linha usando o argumento *type*=:

Script para plotar duas séries temporais em gráfico de linha com legenda

```
plot(ano, tri1, type="b", main="Taxa de ocupação por trimestre dos hotéis -  
Município do Rio de Janeiro", xlab="ano", ylab="Taxa de ocupação %",  
col="blue", ylim=c(50,80))
```

```
lines(ano, tri2, col="red", type="b")
```

```
legend(2007, 80, c("1o.trim","2o.trim"), col=c("blue","red"), pch=rep(20,2))
```

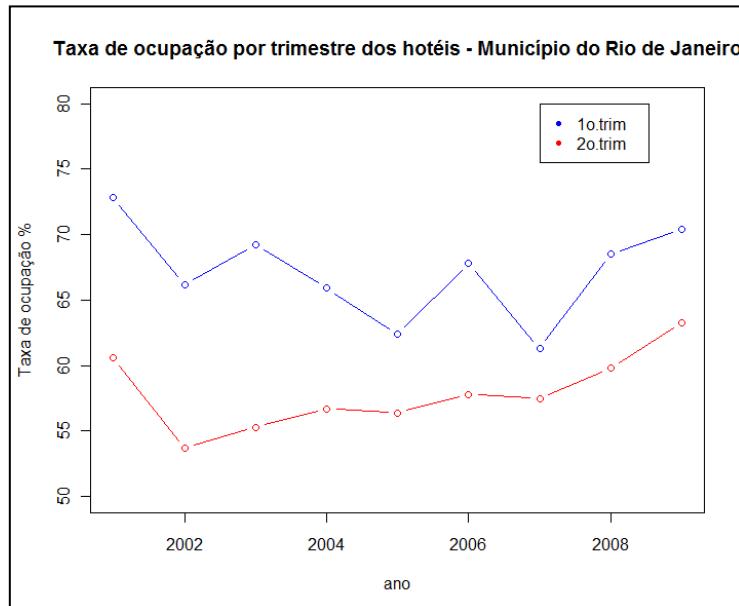


Figura 2-31: Gráfico de linha e pontos de duas séries temporais com legenda

2.9.1 Hora de Praticar

1. Os dados (já retirados do R) mostram a velocidade em que o carro está e a distância que ele demora a parar. Crie um gráfico de linhas, colocando cor na linha, título e nome nos eixos.

cars

attach(cars)

2. Os seguintes dados são medidas de viscosidade para um produto químico observado de hora em hora (leia de cima para baixo e da esquerda para a direita).

47,9 48,8 48,6

47,9 48,1 48,0

48,6 48,3 47,9

48,0 47,2 48,3

48,4 48,9 48,5

48,1 48,6 48,1

48,0 48,0 48,0

48,6 47,5 48,3

Construa um gráfico de série temporal para esses dados.

3. De acordo com Ministério da Educação a quantidade e alunos matriculados no ensino de 1º grau no Brasil nos de 1990 a 1996 em milhares de alunos, são: 19.720 – 20.567 – 21.473 – 21.887 – 20.598 – 22.473 – 23.564. Faça um gráfico de série temporal para apresentar esses dados.

2.10 Gráfico da distribuição normal

A distribuição normal é uma das mais importantes distribuições da estatística, conhecida também como Distribuição de Gauss ou Gaussiana. É inteiramente descrita por seus parâmetros de média e desvio padrão, ou seja, conhecendo-se estes consegue-se determinar qualquer probabilidade em uma distribuição Normal.

Para gerar o gráfico da distribuição de densidade de probabilidade, utilize:

Script para o gráfico da Distribuição Normal

```
curve(dnorm(x), xlim=c(-4,4), main="Distribuição Normal Padrão",
      ylab="Densidade")
```

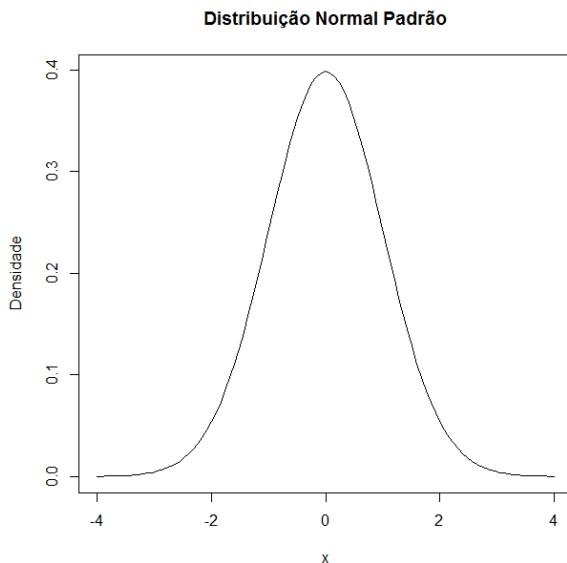


Figura 2-32: Gráfico da distribuição normal de média 0 e desvio padrão 1.

A geração de números aleatórios para a distribuição normal é obtida pelo comando `rnorm(valor)`, sendo valor um número inteiro representando a quantidade de números que se deseja gerar.

```
rnorm(50, mean=2, sd=sqrt(3)) # amostra de 50 número extraídos de uma distribuição normal com média = 2, variância = 3.  
rnorm(20) # amostra de tamanho 20 da distribuição normal padrão, média = 0 e variância =1.
```

Atente para o fato de que cada vez que se roda os comandos acima, uma nova sequência é obtida. Isso pode causar alguma inconveniência quando se deseja compartilhar

o código ou rodá-lo outra vez de forma a obter os mesmos resultados. Para contornar este problema, basta definir um código de semente com o comando `set.seed()` antes de gerar a sequência aleatória.

```
set.seed(1112); rnorm(4)
```

Com a semente 1112, o resultado será sempre 0.78768604, -0.42202478, 0.05690217, 0.71057431.

Vamos produzir um histograma da frequência relativa de uma amostra de 500 números extraídos de uma distribuição normal com média = 2, variância = 4.

```
set.seed(1112); hist(rnorm(500, mean=2, sd=sqrt(4)), freq=FALSE)
```

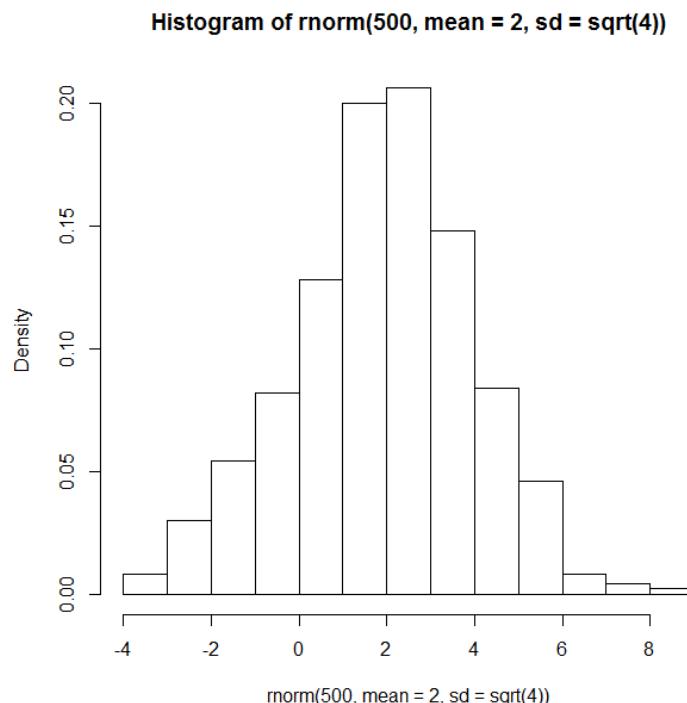


Figura 2-33: Histograma de uma variável com distribuição normal

Podemos traçar no mesmo gráfico a curva da função de densidade da distribuição normal com média 2 e variância 4.

Observe que `col=2` fará o traçado vermelho, `lty=2` fará linha tracejada, `lwd=2` refere-se a espessura da linha e `add=TRUE` sobrepõe a curva ao gráfico anterior.

```
set.seed(1112); curve(dnorm(x, mean=2, sd=sqrt(4)), col=2, lty=2, lwd=2, add=TRUE)
```

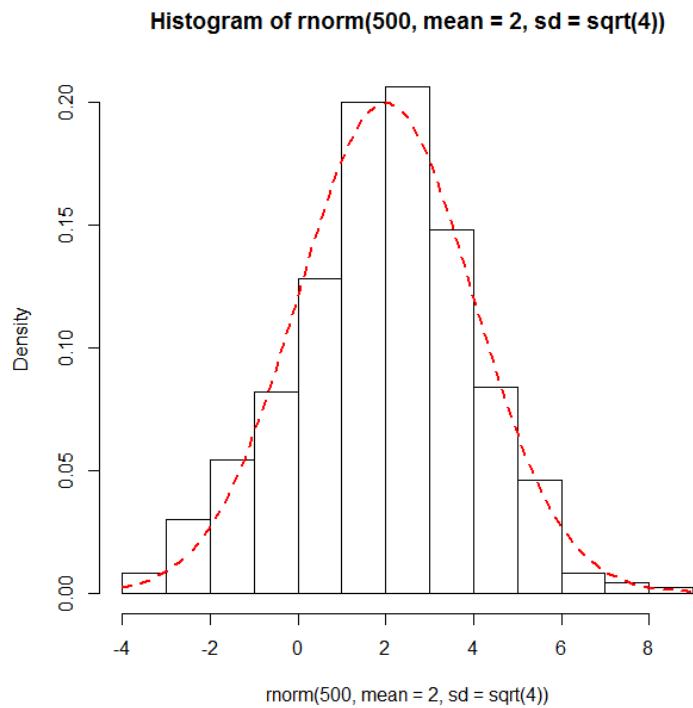


Figura 2-34: Gráfico de histograma com a curva da normal

Vamos ilustrar graficamente o que acontece com os parâmetros da função de densidade normal.

Fixando a variância e variando a média:

```
curve(dnorm(x, mean=1, sd=sqrt(2)), lwd=2, from=-6, to=17)
curve(dnorm(x, mean=10, sd=sqrt(2)), col=2, lwd=2, add=T)
# Adicionando legenda e título ao gráfico:
legend('topright', legend=c(expression(mu==1), expression(mu==10)), text.col=c(1,2),
cex=1.5)
title("Comparando a distribuição normal com médias diferentes")
```

Comparando a distribuição normal com médias diferentes

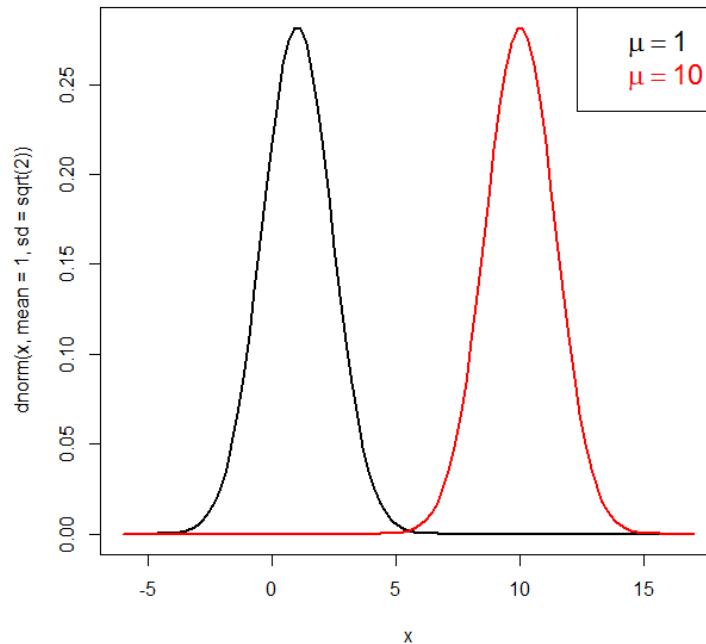


Figura 2-35: Gráfico de duas normais com médias diferentes e legenda

Fixando a média e variando a variância:

```
curve(dnorm(x,mean=4, sd=sqrt(3)), lwd=2, from=-15, to=25)
curve(dnorm(x, mean=4, sd=sqrt(20)), col=2, lwd=2, add=T)
legend('topright', legend=c(expression(sigma^2==3), expression(sigma^2==20)),
text.col=c(1,2), cex=1.5)
title("Comparando a distribuição normal com variâncias diferentes")
```

Comparando a distribuição normal com variâncias diferentes

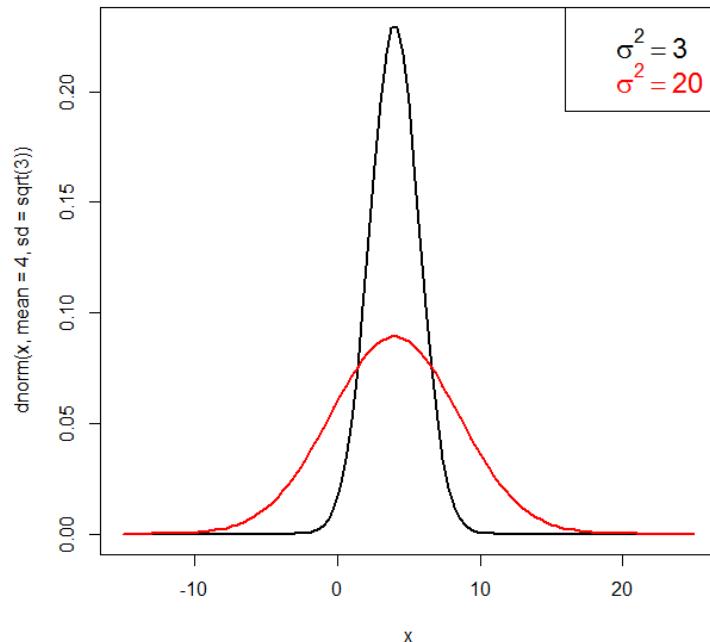


Figura 2-36: Gráfico de duas normais com variâncias diferentes

2.10.1 Hora de Praticar

- 1- Dez observações do tempo efetivo de vida de um catalisador usado em reações químicas produziram os resultados: 1176, 1191, 1214, 1220, 1205, 1192, 1201, 1190, 1183 e 1185. Supondo que estes tempos sigam a distribuição normal, calcule a média e o desvio padrão para construir o gráfico de uma normal com estes parâmetros.
- 2- Construa o gráfico da distribuição normal variando a média no intervalo [-1,1] com tamanho de passo igual a 0.5. Fixe o desvio padrão em 1. Apresente uma legenda e cores diferentes para cada curva.
- 3- Construa o gráfico da distribuição normal variando o desvio padrão no intervalo [0,2] com tamanho de passo igual a 0.5. Fixe a média em 0. Apresente uma legenda e cores diferentes para cada curva.
- 4- Considere o peso de uma amostra de 15 pacotes de açúcar da marca X.

```
pesos=
c(0.9475,0.9705,0.9770,0.9775,0.9860,0.9960,0.9965,0.9975,1.0050,1.0075,1.0100,1.0175
,1.0180,1.02000,1.0250)
```

Construa o histograma destes pesos e no mesmo gráfico insira a curva de uma distribuição normal com média 1 e desvio padrão 0.05 na cor azul e outra com média igual a média dos pesos e desvio padrão igual ao desvio padrão dos pesos na cor vermelha. Coloque legenda para cada curva.

2.11 Referência Bibliográfica do capítulo 2

Alcoforado, L.F., Cavalcante, C.V., **Introdução ao R utilizando a Estatística Básica**, EDUFF, Niterói, 2014.

Arriaza Gómez, A.J. et al. **Estadística Básica con R y R-commander**. Version Febrero 2008. Servicio de Publicaciones de La Universidad de Cadiz, 2008.

Crawley, M.J. **The R Book**. Ed. John Wiley Professio, 2007.

IBGE. **Dados sobre Niterói**, disponível em <http://www.ibge.gov.br/cidadesat/>. Acesso em 08/12/11.

Morettin, P.A.; Bussab, W.O. **Estatística Básica**. 5.ed. Ed. Saraiva, 2009.

R Development Core Team, **R: A Language and Environment for Statistical Computing**,**R Foundation for Statistical Computing**, Vienna, Austria, 2009, disponível em <http://www.R-project.org>. Acesso em 08/12/15.

3 Gráficos usando pacote *Lattice*

Ariel Levy

Embora muito tenha sido visto no pacote anterior, o R ainda possui outras possibilidades. Esta é uma das vantagens de utilizar-se *software* livre. Diversos autores propõem diferentes soluções em pacotes que apresentam maior ou menor adequabilidade as tarefas do analista de dados.

O propósito de visualizar dados é comunicar e o pacote *Lattice*, desenvolvido por Deepayan Sarkar, tem por característica principal aprimorar a visualização de relacionamentos multivariados. Em particular, o pacote suporta a representação de gráficos em treliça (*trellis graphs*) – que expõe o relacionamento entre uma ou mais variáveis quando condicionadas a uma ou mais variáveis.

O *Lattice* requer a escolha de um tipo de gráfico, uma fórmula e um conjunto de dados. Difere do pacote básico sendo todas as instruções passadas de uma única vez. Assim, para alterar o gráfico será preciso executar novamente.

Tipo_do_gráfico (fórmula, data= [arquivo])

Assim, o analista deverá escolher inicialmente o tipo do gráfico desejado dentre os quinze disponíveis no pacote. Em seguida a fórmula desejada e por fim o conjunto de dados.

A Tabela 3.1 apresenta os tipos de gráficos disponibilizados neste pacote.

As funções (~) são apresentadas de forma simplificada onde x, y ou z representam variáveis aleatórias e C e F variáveis condicionais do tipo *factor* e *data frame* um conjunto de dados. Então, ~x|C apresenta a variável numérica x para cada nível da variável categórica, C, do tipo *factor* no R. Enquanto y~x| C*F apresenta a relação entre as duas variáveis y e x para cada combinação das variáveis categóricas C e F.

Tabela 3.1: Tipos de gráfico no Lattice – adaptado de Kobakof (2014) e Sakar (2015)

Tipo de Gráfico	Descrição	Exemplo de fórmula
barchart	gráfico de barras	$x \sim C$ ou $C \sim x$
bwplot	boxplot	$x \sim C$ or $C \sim x$
cloud	gráfico de dispersão em 3D	$z \sim x^*y C$
contourplot	gráfico de contorno em 3D	$z \sim x^*y$
densityplot	gráfico da densidade de Kernel	$\sim x C^*F$
dotplot	gráfico de pontos empilhados	$\sim x C$
histogram	histograma	$\sim x$
levelplot	gráfico de curvas de níveis em 3D	$z \sim y^*x$
parallel	gráfico de coordenadas paralelas	data frame
qq()	gráfico de quartis para duas amostras	$x \sim y C$
qqmath()	gráfico do quartil teórico	$\sim x C$
splom	matriz de dispersão	data frame
stripplot	gráficos em tiras	$C \sim x$ ou $x \sim C$
xyplot	gráficos de dispersão	$y \sim x C$
wireframe	graficos de superfície 3D	$z \sim y^*x$

3.1 Gráfico de Dispersão no Lattice

Nosso primeiro exemplo será um simples gráfico de dispersão, utilizando a base de dados *mtcars* disponibilizado pelo R.

```
library(lattice)
xyplot(mpg~wt, data=mtcars, pch=19, main="MPG vs WT", xlab="Wt/1000",
       ylab="MPG")
```

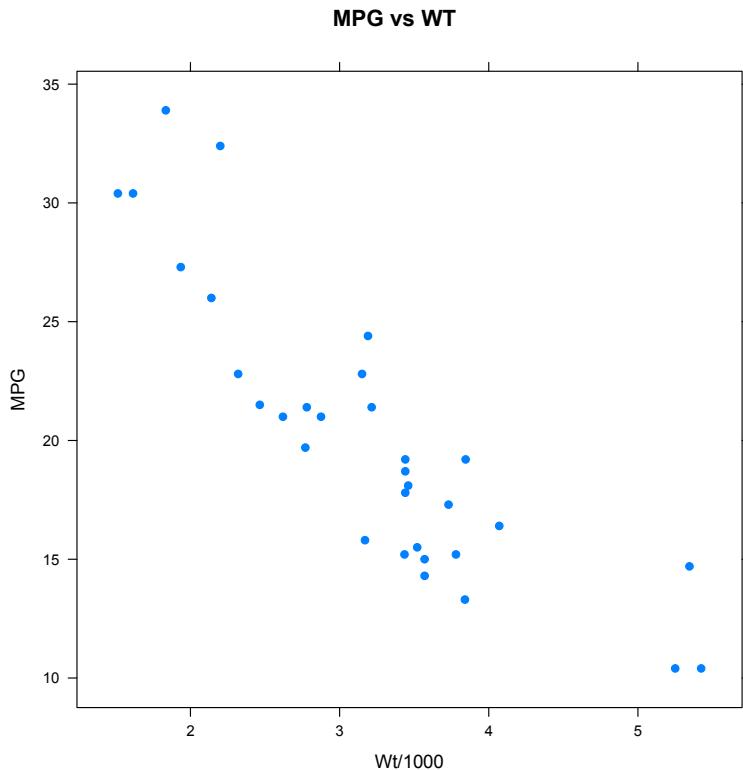


Figura 3-1: Gráfico de Dispersão – dados MTCARS pacote Lattice.

Na primeira linha carregamos o pacote *Lattice* e na segunda criamos o gráfico de dispersão mostrado na Figura 3-1. Assim, como no pacote básico é possível escolher o símbolo, colocar o título e dar nome aos eixos. Mas não exploramos a principal característica deste pacote. Com uma pequena alteração poderemos condicionar o consumo e peso ao número de cilindros.

```
xyplot(mpg~wt | factor(cyl), data=mtcars, pch=19, main="MPG vs Wt",
       xlab="Wt/1,000", ylab="MPG", layout=c(3,1), type=c("p","g"))
```

O resultado mostrado na Figura 3-2 é bem satisfatório considerando que utilizou-se apenas uma linha de código.

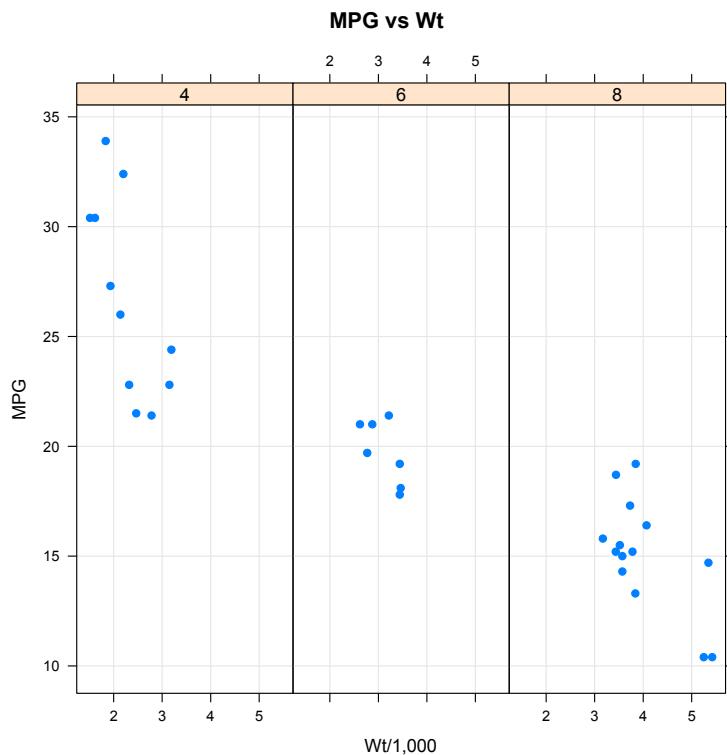


Figura 3-2: Gráfico de dispersão condicionado ao número de cilindros.

Algumas instruções foram adicionadas: a variável categórica – número de cilindros entrou na fórmula, o *layout* especificando que teríamos 3 painéis em uma linha. Vejam que conhecíamos de antemão o número de diferentes categorias de cilindros. Caso não soubéssemos poderíamos ter utilizado o comando *unique(mtcars\$cyl)* em substituição ao número 3 diretamente no comando. O *type* ajusta o uso da grade no gráfico, se suprimido esta seria eliminada.

Mas isto foi apenas para dar noção de quão poderoso é este pacote. Fica ainda melhor se utilizarmos simultaneamente duas variáveis categóricas condicionando duplamente o resultado. Incluiremos na fórmula a variável categórica relativa a transmissão, *am* para a qual definimos os rótulos como “A” e “M” correspondendo respectivamente a automática ou manual.

```
xyplot(mpg~wt | factor(cyl) + factor(am,labels=c("A","M")), data=mtcars,
main="MPG vs Wt", xlab="Wt/1,000", ylab="MPG", pch=19, type=c("p","g"))
```

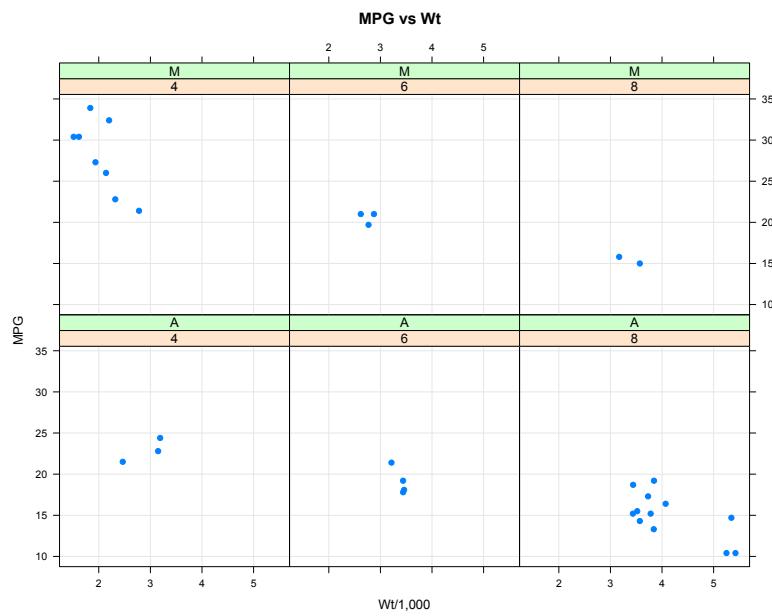


Figura 3-3: Gráfico de dispersão duplamente condicionado.

Agora foi deixado ao *Lattice* resolver como apresentar o resultado. O consumo e peso dos veículos foi distribuído em duas linhas conforme a transmissão nas colunas como antes o número de cilindros. Um gráfico parecido com o primeiro (Figura 3-1) utilizando o pacote básico demandaria uma programação bem mais complicada. Enquanto que um parecido a este seria mais fácil pois pacote básico lida bem com gráficos X/Y, cuja instrução seria: **coplot(mpg~wt|factor(cyl)+factor(am),data=mtcars)** embora o resultado apresentado na Figura 3-4 estivesse bem aquém.

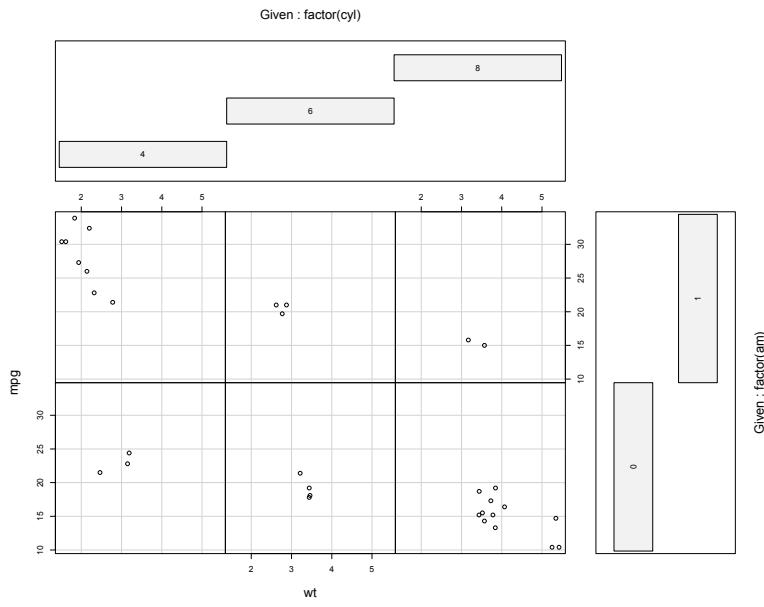


Figura 3-4: Gráfico de dispersão duplamente condicionado com o pacote básico.

3.2 Gráfico Stripplot

O *stripplot* pode ser um gráfico interessante para revelar aspectos de uma distribuição especialmente com poucos dados, vejamos o exemplo disposto no *vignette* do pacote *Lattice*.

```
stripplot(depth ~ factor(mag), data = quakes, jitter.data = TRUE, alpha = 0.6, main =
  "Depth of earthquake epicenters by magnitude", xlab = "Magnitude (Richter)", ylab =
  "Depth (km)")
```

Note como a transparência parcial e o efeito *jitter* aliviam a superposição dos pontos neste gráfico em tiras.

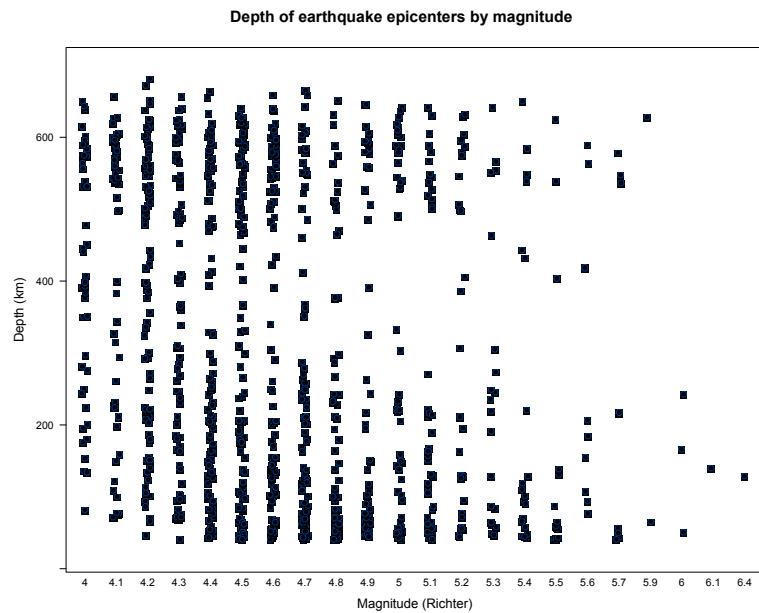


Figura 3-5: Stripplot exemplo de gráfico de tiras – Sarkar (2015)

3.3 Gráfico em 3D

Outra possibilidade no uso deste pacote são gráficos em 3D como o *wireframe*, a título de exemplo iremos gerar os dados para apresentar uma normal bivariada com este comando. Iniciamos por gerar as sequências e formar a grade uniforme. Em seguida calculamos os valores para uma função z, e finalmente apresentamos o gráfico, lembre-se que o pacote já se encontra carregado.

```
x<-y<-seq(-5, 5, length=50)
xy<-expand.grid(x, y)
z<-(1/(2*pi))*exp(-.5*(xy[,1]^2+xy[,2]^2))
wireframe(z~xy[,1]*xy[,2], xlab="x", ylab="y")
```

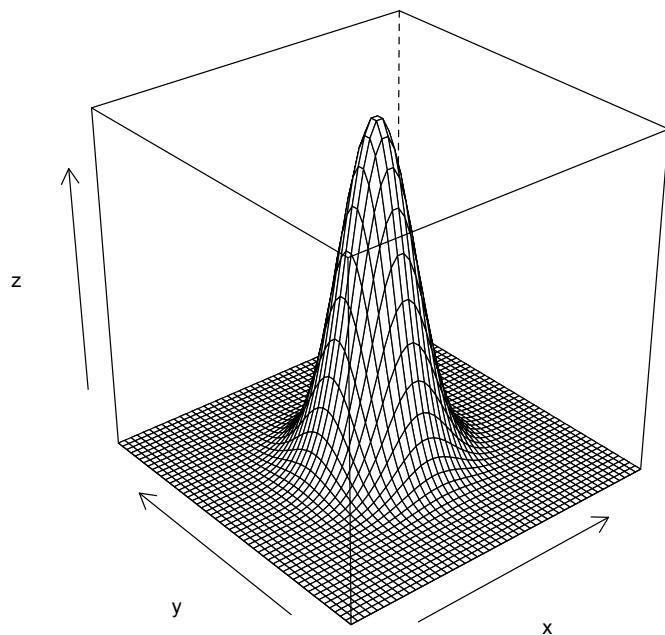


Figura 3-6: Wireframe gráfico em 3D

Com isso cremos ter demonstrado ao leitor que cada pacote no R tem características e finalidades próprias e embora não tenhamos esgotado as possibilidades com o *Lattice* esperamos ter mostrado como ele pode ser útil.

3.4 Referências Bibliográficas do Capítulo 3

PITTARD, Steve. **Conditioning and Grouping with Lattice Graphics**. Disponível em <https://rollingyours.wordpress.com/2014/02/17/conditioning-and-grouping-with-lattice-graphics/>. Acesso em 13/12/2015.

RIZZO, Maria L. **Statistical Computing with R**, Chapman & Hall/CRC, Florida Boca Raton, 2008.

SAKAR, Deepayan **Lattice Multivariate Data Visualization with R**, Use R! Series, Springer , Baltimore 2011.

SAKAR, Deepayan, **Package lattice**. Disponível em <https://cran.r-project.org/web/packages/lattice/lattice.pdf>. Acesso em 13/12/2015.

4 Gráficos usando pacote *ggplot2*

Luciane Ferreira Alcoforado

O *ggplot* é um pacote do Software R que é utilizado para a criação de gráficos. Como todo pacote, é necessária a instalação, que é muito simples. Basta digitar

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

e antes de utilizá-lo, lembrar de carregá-lo:

```
library(ggplot2).
```

Também é necessário que a versão do R seja recente. Então se você não tem certeza sobre isso, visite a página <https://www.r-project.org/> para ver as novidades.

O pacote foi idealizado por Hadley Wickham e Winston Chang que desenvolveram a ideia de construir diversos tipos de gráficos a partir da mesma estrutura de componentes: *data*, referente ao banco de dados; *geom_forma*, com um rol de tipos possíveis de representação dos dados (veja Tabela 4.1) e *coord_system*, referente ao sistema de coordenadas, que pode ser cartesiana, polar e projeção de mapas.

O gráfico feito com o *ggplot* é construído a partir de camadas (*layers*), cada camada é sobreposta uma após a outra gerando o desenho final. Existem comandos específicos para cada camada, mas informação é possível obter no site <http://docs.ggplot2.org>, na língua inglesa.

Para melhor entendimento, será utilizado os dados do capítulo 2, proporcionando uma comparação entre o modo “básico” e o *ggplot*. Cada acréscimo de camada é feito com um sinal de “+”. A primeira camada é sempre a informação básica, ou seja, os dados. A próxima camada é o tipo de gráfico que se quer criar. A partir disso, depende da pessoa o que ela quer colocar em seu gráfico; título, eixos, cor, etc. E também é possível salvar o *plot* com o argumento “*ggsave*”.

O esquema para todos os tipos de gráficos disponíveis neste pacote é:

```
ggplot(banco de dados, aes(x = variavel)) + geom_forma()
```

O banco de dados assumiremos que esteja depositado em arquivo BD.xls, representando os dados de uma tabela. Das possibilidades de *forma* do comando do *ggplot2* destacamos as mais comuns na tabela 4.1:

Tabela 4.1: Possibilidades de forma para gráficos no ggplot2

Forma	Tipo de gráfico
bar	barras
Coord_polar	setores
histogram	histograma
boxplot	boxplot
point	dispersão
line	linha
abline	reta

Para fazemos modificações nos detalhes das legendas, cores, medidas, etc, usa-se:

labs() – Para modificar título e nome dos eixos

coord_flip() – Para inverter a posição do eixo x e y

scale_x_continuous – Para modificar o eixo x

Por último, se quiser salvar a imagem basta acrescentar *ggsave(gráfico, file='nome.pdf')* e este não deve ser seguido pelo sinal +. É possível modificar a largura e a altura da figura com os argumentos *width* e *height*. Por exemplo,

```
ggsave(gráfico, file='nome.pdf', width = 10, height = 6)
```

4.1 Gráfico de Barras no ggplot2

Vamos utilizar os dados da Tabela 2.1 do capítulo 2, para mostrar como podemos gerar os mesmos tipos de gráficos da seção 2.4. Considere que estes dados estejam no arquivo BD.csv, conforme pode ser visto na Tabela 4.2 que mostra também como deve ser a arquitetura do banco de dados para os *scripts* apresentados na sequência.

Tabela 4.2: Arquitetura do Banco de Dados para gráfico de barras ggplot

Área	Frequência	Instituição
Saúde	1751	A
Exatas	2186	A
Humanas	947	A
Comunicação	29	A
Saúde	2528	B
Exatas	2132	B
Humanas	1843	B
Comunicação	280	B

Atente para a arquitetura do banco de dados, que no nosso caso será formado por três colunas. Veja que é uma outra forma de representar os mesmos dados da tabela 2.1.

Em informações básicas, x = vetor de dados do eixo x ; y = vetor de dados para frequência no eixo y e $fill$ representa as categorias de contagem.

No argumento *geom_bar*, uma vez que as frequências já estão consolidadas, deve-se usar o argumento *stat= "identity"*, caso contrário use *stat= "bin"*. Quanto ao argumento *width*, refere-se a largura das barras. Quanto ao argumento *position* temos a opção *dodge* (lado a lado); ou *fill* (empilhado).

No argumento *labs* informamos os títulos e rótulos dos eixos.

Para o tema geral do gráfico há as seguintes opções:

theme_bw() – possui fundo branco e linhas de grade

theme_gray() – possui fundo cinza e linhas de grade

theme_classic() – possui fundo branco sem linhas de grade

theme_minimal() – tema mínimo

Script para gráfico de Barra

```
dados <- read.csv2("C:\\...\\BD.csv")
p<-ggplot(data = dados, aes(x=Área, y=Frequênciа, fill=Instituição))+ # Informação
Básica
  geom_bar(stat='identity', width=0.5, position = "dodge") +
  labs(x='Área', y='Frequênciа', title="Distribuição de Bolsas por Área do
Conhecimento") +
  theme_bw()
p #plota o gráfico na janela gráfica
#gravando o gráfico em arquivo pdf do diretório corrente:
ggsave(p, file='grafico.pdf', width = 10,height = 6)
```

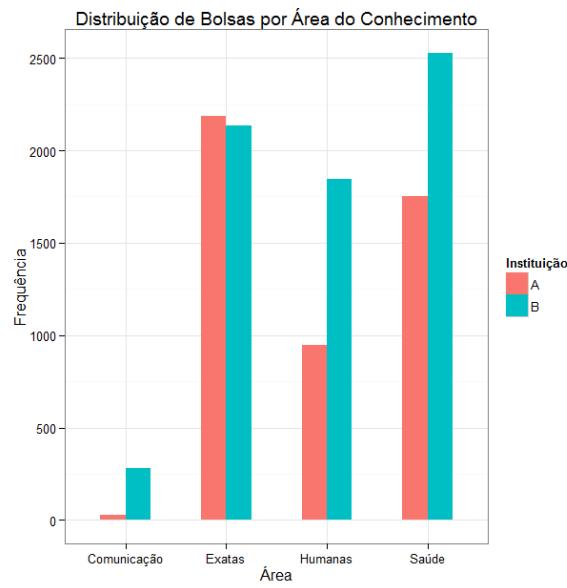


Figura 4-1: Gráficos de Barras lado a lado no ggplot

Script para gráfico de Barra Empilhado

```
p<-ggplot(data = dados, aes(x=Área, y=Frequênci, fill=Instituiçāo))+ # Informaçāo
Básica

geom_bar(stat='identity', width=0.5, position = "fill") +
labs(x='Área' , y='Frequênci, title="Distribuiçāo de Bolsas por Área do
Conhecimento") +
theme_gray()

p #plota o gráfico na janela gráfica+theme_gray()
```

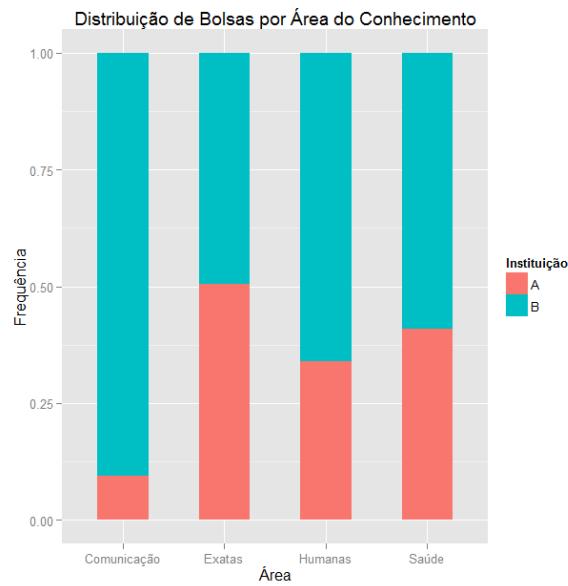


Figura 4-2: Gráficos de Barras empilhado no ggplot

Vamos agora fazer o gráfico preenchendo as colunas por área

```
p<-ggplot(data = dados, aes(x=Instituição, y=Frequência, fill=Área))+ # Informação
Básica
  geom_bar(stat='identity', width=0.5, position = "fill") +
  labs(x='Instituição', y='Frequência', title="Distribuição de Bolsas por Instituição")+
  theme_gray()
p #plota o gráfico na janela gráfica+theme_gray()
```

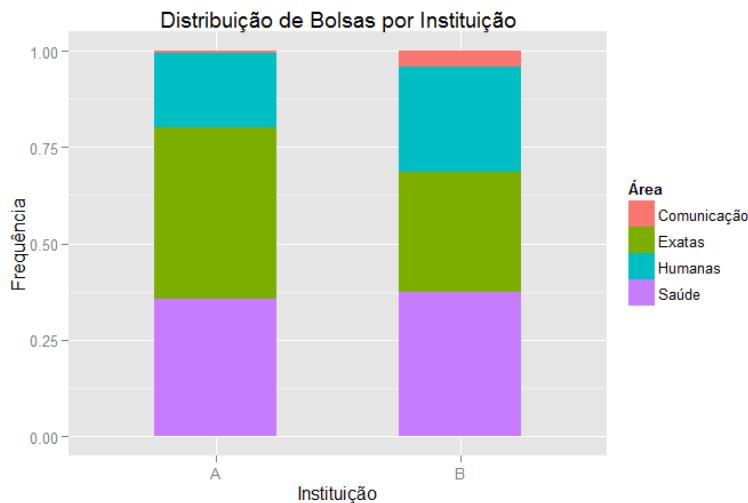


Figura 4-3: Gráficos de Barras empilhado no ggplot, fundo cinza, frequência relativa
`p<-ggplot(data = dados, aes(x=Instituição, y=Frequência, fill=Área)) +`

```
geom_bar(stat='identity', width=0.5) +
labs(x='Instituição', y='Frequência', title="Distribuição de Bolsas por Instituição")+
theme_bw()
```

`p`

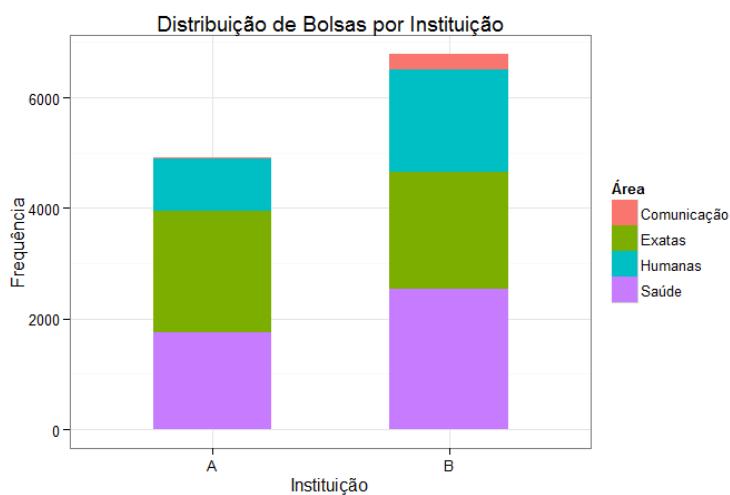


Figura 4-4: Gráficos de Barras empilhado no ggplot, fundo branco, frequência absoluta

4.1.1 Hora de Praticar

- 1- Desenvolva o gráfico da figura 4.3 e 4.4 na versão lado a lado.
- 2- Desenvolva os gráficos para os casos da seção 2.4.7

4.2 Gráfico de Setor no ggplot2

Observe o gráfico da figura 4.3. O script abaixo possibilita converter este gráfico de barras em um gráfico de setores. O grande diferencial neste recurso é produzir um único gráfico para duas sequências, a da instituição A e a da instituição B, tudo isso na forma de anéis, o interno representando A e o externo B, permitindo uma comparação imediata das duas categorias.

Script para gerar um gráfico de setores

```
p<-ggplot(data = dados, aes(x=Instituição, y=Frequência, fill=Área))+  
  geom_bar(stat='identity', width=0.5, position = "fill") +  
  labs(x='Instituição', y='Frequência', title="Distribuição de Bolsas por Instituição") +  
  theme_gray()  
  
p + coord_polar(theta="y")
```

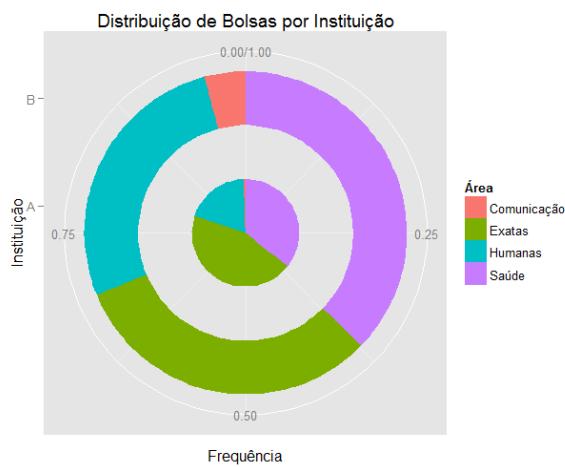


Figura 4-5: Gráficos de Setores no ggplot

4.3 Histograma no ggplot2

Usaremos os dados do *mtcars*. Veja como produzir um histograma padrão do pacote usando a camada `+geom_histogram()`.

```
ggplot(data=mtcars, aes(mpg)) + geom_histogram()
```

Vamos aprimorar acrescentando alguns elementos como amplitude das colunas (*breaks*) e cores (*color*). Veja o resultado na figura abaixo.

Script para histograma no ggplot2

```
h=ggplot(data=mtcars, aes(mpg)) + geom_histogram(breaks=seq(10, 35, by = 5),  
color="white", fill= "black")
```

```
h
```

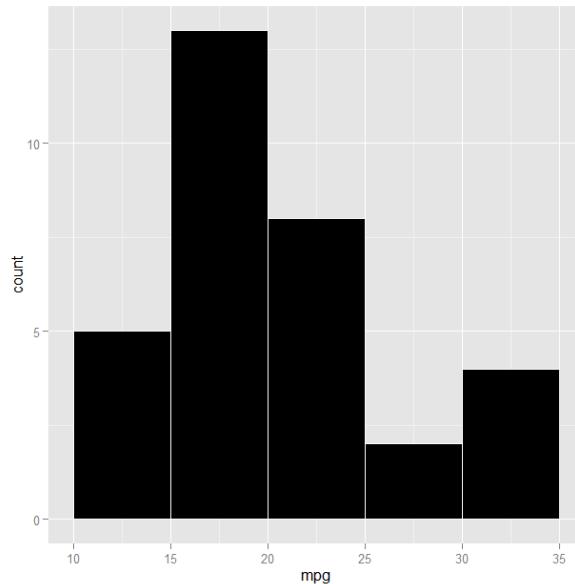


Figura 4-6: Histograma no ggplot

Podemos ainda ajustar as cores de preenchimento das colunas (*fill*) de acordo com os valores de suas frequências, desse modo temos a visualização de comparação não só pela altura das colunas como também pelo termômetro de cores de cada uma.

Script de histograma no ggplot com termômetro de cores

```
h=ggplot(data=mtcars, aes(mpg)) + geom_histogram(breaks=seq(10, 35, by = 5),  
color="white", aes(fill=..count..))
```

```
h
```

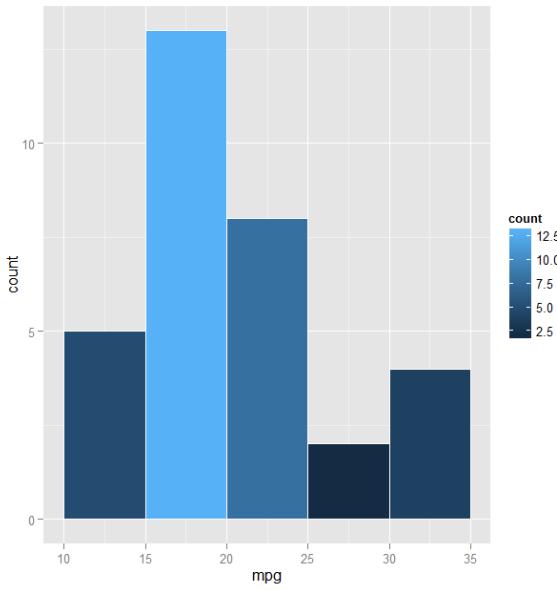


Figura 4-7: Histograma no ggplot com termômetro de cores para as contagens.

Script de histograma no ggplot com termômetro de cores e linha de tendência

```
h=ggplot(data=mtcars, aes(mpg) ) + geom_histogram(aes(y= ..density.., fill=..count..),
breaks=seq(10, 35, by = 5), color="white") +geom_density(col=2) )
h
```

4.4 BoxPlot no ggplot2

Usaremos os mesmos dados da seção 2.7, a amostra de 48 pedras de uma reserva de petróleo.

```
> rock[1:6,]
  area      peri      shape perm
1 4990 2791.90 0.0903296  6.3
2 7002 3892.60 0.1486220  6.3
3 7558 3930.66 0.1833120  6.3
4 7352 3869.32 0.1170630  6.3
5 7943 3948.54 0.1224170 17.1
6 7979 4010.15 0.1670450 17.1
```

No comando *ggplot* temos sempre que definir a variável x e y. O *boxplot* será da variável *shape* podendo ser observado a amplitude da variável *perímetro*, representada no eixo x.

Script para BoxPlot simples

```
b<-ggplot(data = rock, aes(x=peri, y=shape)) + geom_boxplot(color= "Blue")
```

b

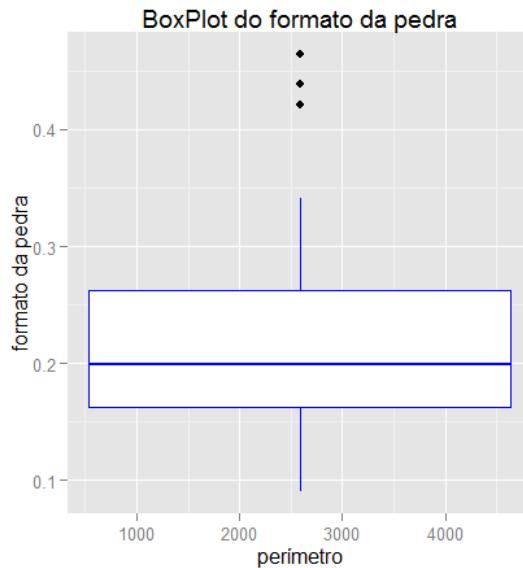


Figura 4-8: Gráficos BoxPlot com alteração de cor e inclusão de títulos
Para acrescentar título, novos rótulos para os eixos:

```
b +ggtitle("BoxPlot do formato da pedra") + xlab("perímetro") + ylab("formato da pedra")
```

Se quiser modificar a cor utilize `color= "nome da cor"` ou número

```
b<-ggplot(data = rock, aes(x=peri, y=shape)) + geom_boxplot(color= "Blue")
```

ou

```
b<-ggplot(data = rock, aes(x=peri, y=shape)) + geom_boxplot(color= 3)
```

Para múltiplos gráficos, utilizaremos os dados de *PlantGrowth*

```
PlantGrowth[1:6,]  
  weight group  
1   4.17  ctrl  
2   5.58  ctrl  
3   5.18  ctrl  
4   6.11  ctrl  
5   4.50  ctrl  
6   4.61  ctrl
```

Temos duas possibilidades para gerar o *boxplot* da variável *weight*:

```
b<-ggplot(data = PlantGrowth, aes(x=group, y=weight)) + geom_boxplot()
```

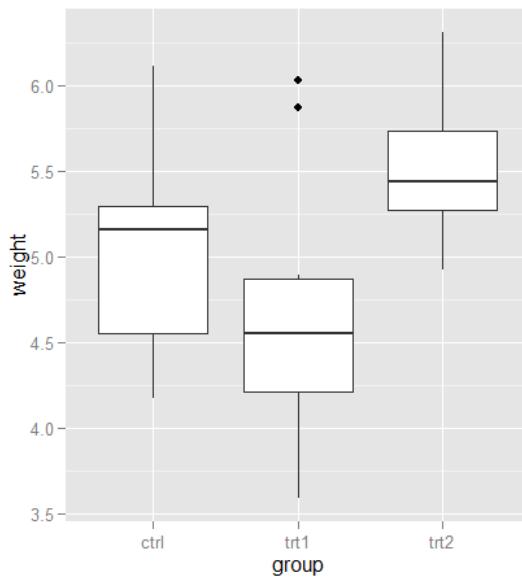


Figura 4-9: Gráficos BoxPlot com vários grupos

```
b<-ggplot(data = PlantGrowth, aes(x=group, y=weight, fill=group)) + geom_boxplot()
```

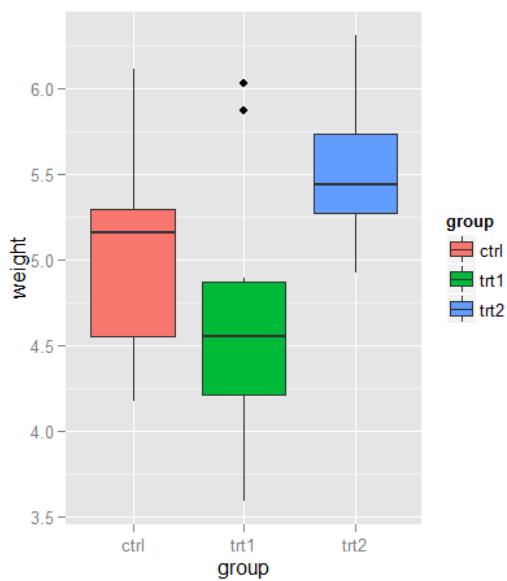


Figura 4-10: Gráficos BoxPlot com vários grupos, legenda e cores

Para colocar cor no *outlier* o comando é *outlier.colour = "cor"*. Nesse gráfico também foram colocados pontos que representam como os dados estão dispersos, com o comando *geom_jitter*. Você também pode modificar esses pontos, como o tamanho (*size*) e cor (*colour*).

```
b<-ggplot(data = PlantGrowth, aes(x=group, y=weight, fill=group)) +
geom_boxplot(outlier.colour= "red") +
geom_jitter(size=2, colour= "blue")
```

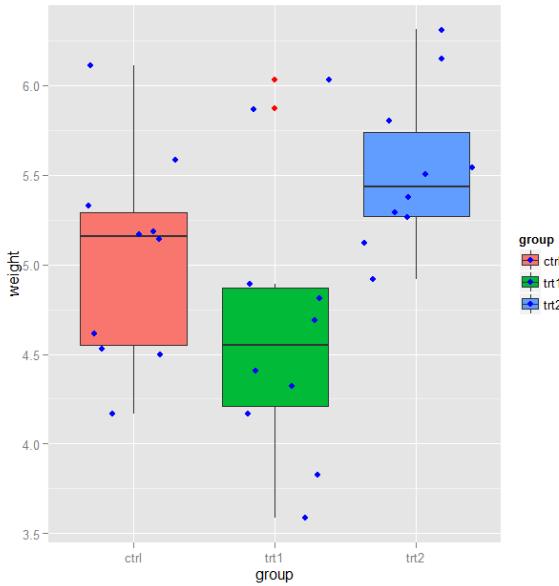


Figura 4-11: Gráficos BoxPlot com vários grupos incluindo dispersão dos dados

Invertendo a posição do gráfico.

```
b<-ggplot(data = PlantGrowth, aes(x=group, y=weight, fill=group)) +
geom_boxplot(outlier.colour= "red") +
geom_jitter(size=2, colour= "blue") +
coord_flip()
```

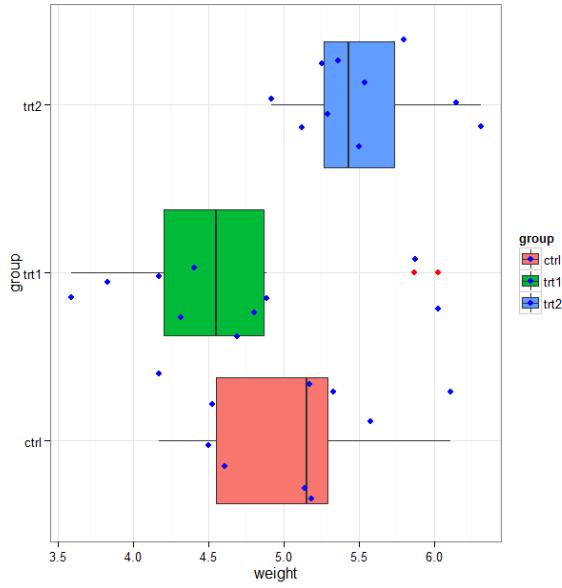


Figura 4-12: Gráficos BoxPlot com vários grupos incluindo dispersão na posição horizontal

Em qualquer dos casos, os títulos são acrescentados sempre da mesma forma:

```
b +ggtitle("Boxplot para rendimento de plantas segundo o tratamento ") +
  xlab("tratamento") + ylab("peso")
```

4.4.1 Hora de Praticar

Proponha gráficos no *ggplot* para os itens da seção 2.7.1

4.5 Gráfico de dispersão no ggplot2

Utilizaremos o banco de dados *Orange*. Os dados são medidas da circunferência em mm do tronco de pés de laranjeiras e sua respectiva idade em dias.

```
Orange[1:6,]
```

	Tree	age	circumference
1	1	118	30
2	1	484	58
3	1	664	87
4	1	1004	115
5	1	1231	120
6	1	1372	142

```
b<-ggplot(data = Orange, aes(x=age, y=circumference, fill=Tree)) + geom_point()
```

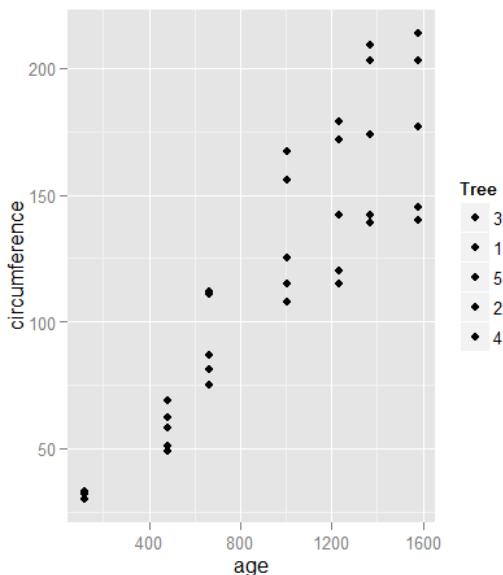


Figura 4-13: Gráficos de Dispersão para 5 espécies de árvores no ggplot

Para discriminar os pontos, usamos um comando de coloração dos pontos de acordo com alguma das variáveis. Isso facilitará a leitura e interpretação dos dados.

```
b + geom_point(aes(colour = Tree))
```

ou

```
b + geom_point(aes(colour = age))
```

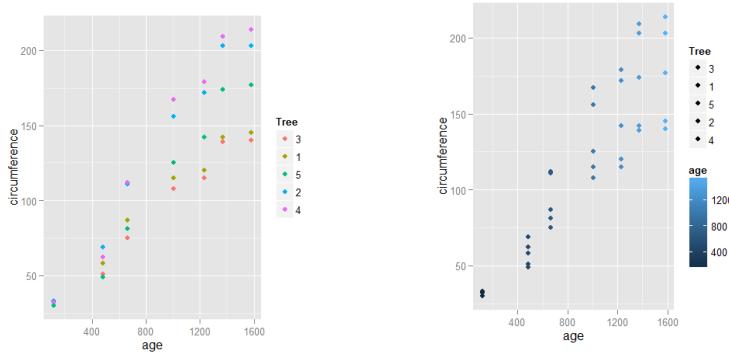


Figura 4-14: Gráficos de Dispersão para 5 espécies de árvores com cores por categoria

Observe que alguns pontos ficaram sobrepostos. Para melhorar a visualização de casos como este, use a forma *jitter*.

```
b<-ggplot(data = Orange, aes(x=age, y=circumference, fill=Tree)) + geom_jitter()
```

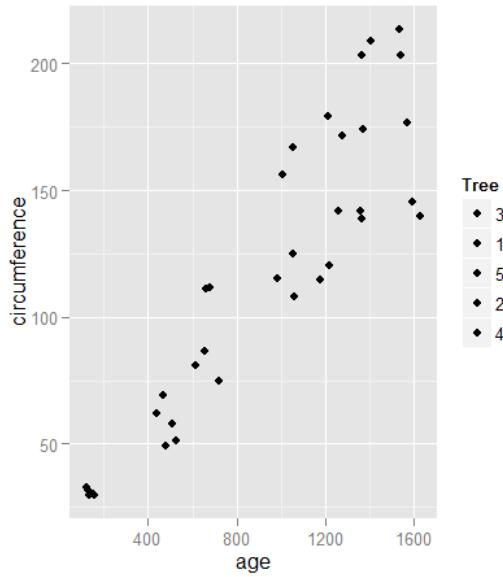


Figura 4-15: Gráficos de Dispersão para 5 espécies de árvores comando jitter

Com este recurso é possível perceber a massa de dados que se concentra num ponto de sobreposição, permitindo uma análise mais precisa.

4.6 Gráfico de linhas no ggplot2

Utilizaremos os dados da seção 2.9. Vamos criar dois grupos para compor o *dataframe*, um representando a função $-x^3$ (grupo 1) e outro a função x^3 (grupo 2). Veja o *script* a seguir.

Script gráfico de linhas de duas funções

```
x <- rep(0:20,2)  
y <- c( -x[1:21]***3, x[1:21]***3)  
dados <- data.frame(x, y, grupo=factor(c(rep(1, 21), rep(2, 21))))  
p=ggplot(dados, aes(x, y, group=grupo, color=factor(group)))  
p + geom_line()
```

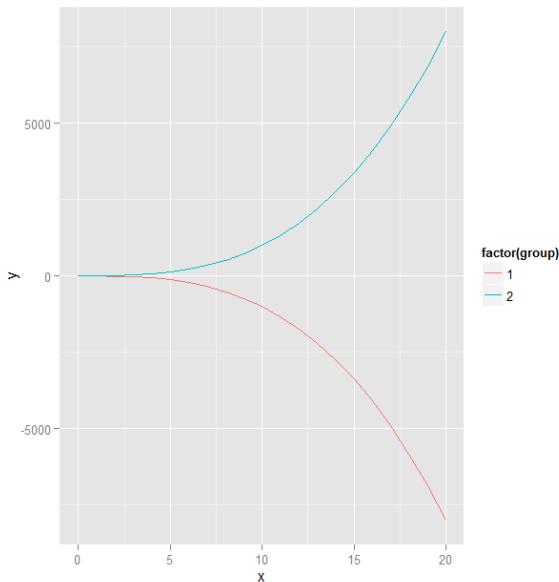


Figura 4-16: Gráficos de Linhas para duas funções no ggplot

Vamos produzir o gráfico para as temperaturas de Ithaca.

```
mes<-1:12  
temperatura<-c(-5.44, -5.17, 0.11, 6.89, 12.67, 17.94, 20.44, 19.5, 15.67, 9.72, 4.06, -2.56)  
dados <- data.frame(mes, temperatura, grupo=factor(c(rep(1, 12))))  
p=ggplot(dados, aes(x=mes, y=temperatura, group=grupo))  
p + geom_line(color=2) #podemos especificar a cor da linha
```

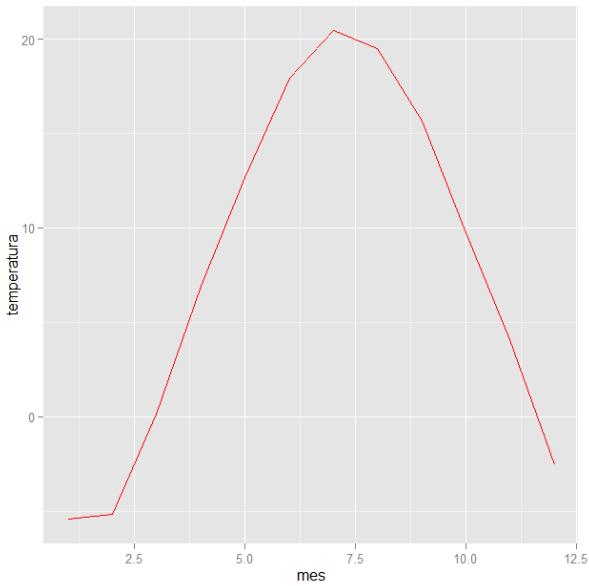


Figura 4-17: Gráficos de Linhas simples no ggplot

Observe, entretanto que há um problema na escala do eixo x. Podemos definir novos limites para melhor apresentação. Utilize uma camada para definir os limites do eixo x.

```
p +geom_line(color=2)+coord_cartesian(xlim=c(0, 12))
```

Para marcar os pontos no gráfico, acrescente mais uma camada (+*geom_point*).

```
p +geom_line(color=2)+coord_cartesian(xlim=c(0, 12))+geom_point(color="black")
```

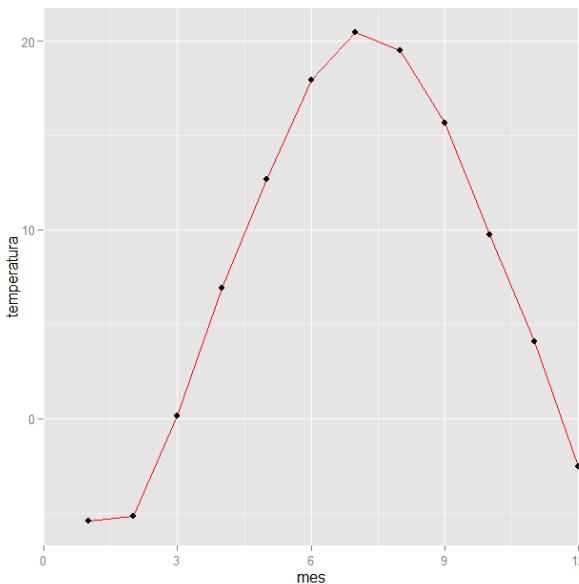


Figura 4-18: Gráficos de Linhas e pontos para a série de temperatura de Ithaca.

4.6.1 Hora de Praticar

Produza um script com o que você aprendeu nesta seção para elaborar um gráfico equivalente ao da Figura 2-31.

4.7 Títulos dos gráficos

Os títulos dos gráficos e eixos são inseridos da mesma forma em todos os tipos de gráfico do pacote *ggplot2*.

Podemos acrescentar camada por camada para cada eixo e Título:

```
+ggtitle("Título do Gráfico")  
+xlab("rótulo do eixo X")  
+ylab("rótulo do eixo Y")
```

Ou todos de uma única vez:

```
+ labs(x="rótulo do eixo X", y="rótulo do eixo Y", title="Título do Gráfico")
```

Não temos como esgotar todas as possibilidades deste pacote, exploramos aqui os principais recursos que tornam viável a produção de visualizações mais utilizadas no meio acadêmico.

4.8 Referência Bibliográfica do capítulo 4

Wickham, Hadley. **ggplot2: Elegant Graphics for Data Analysis**. Dordrecht, Heibelberg, London, New York: Springer, 2009.

_____, **Package ggplot2.** Disponível em <https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>. Acesso em 13/12/2015.

_____, **ggplot2.** Disponível em <http://ggplot2.org/>, 2013. Acesso em 13/12/2015

Rbloggers. **How to make a histogram with-ggplot2** by Filip Schouwenaars. 12/05/2015. Disponível em <http://www.r-bloggers.com/how-to-make-a-histogram-with-ggplot2/>. Acesso em 13/12/2015.

RStudio, **Data Visualization with ggplot2 Cheat Sheet**, 3/2015. Disponível em <https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>. Acesso em 13/12/2015.

5 Gráficos usando pacote *googleVis*

Luciane Ferreira Alcoforado

Em maio de 2010 o *Google* já dava sinais de inovação na forma como os dados poderiam ser representados em gráficos. Era anunciado um conjunto de modelos que possibilitavam a criação de gráficos com animação. Estes modelos foram impulsionados pela apresentação do *TED* de Hans Rosling em 2006 que já ultrapassou a marca de 10 milhões de visualizações.

Atualmente é possível consultar a galeria de gráficos do *google* em <https://developers.google.com/chart/interactive/docs/gallery>.

Em agosto de 2010 Markus Gesmann e Diego de Castillo iniciaram o desenvolvimento do pacote *GoogleVis* que cria uma interface entre o R e o pacote de gráficos do Google. A vantagem em utilizar este pacote é possibilitar ao usuário produzir gráficos diretamente do R, sem precisar carregar os dados para o *google*. Além do poder visual, também possuem animação e podem ser disponibilizados online em sites, blogs ou redes sociais.

5.1 Instalando o pacote

Para instalar o pacote utilize o comando:

```
install.packages( 'googleVis' )
```

Para carregar o pacote use o comando

```
library( googleVis )
```

Neste caso, deve aparecer no console:

```
Welcome to googleVis version 0.5.10
Please read the Google API Terms of Use
before you start using the package:
https://developers.google.com/terms/
Note, the plot method of googleVis will by default use
the standard browser to display its output.
See the googleVis package vignettes for more details,
or visit http://github.com/mages/googleVis.

To suppress this message use:
suppressPackageStartupMessages(library(googleVis))
```

Caso ao final apareça uma mensagem como a que segue,

Warning message:

package ‘googleVis’ was built under R version 3.2.2

significa que sua versão do R é anterior à atual em relação ao pacote. É apropriado que se atualize para a versão 3.2.2. Lembre-se que estas atualizações não são feitas automaticamente.

Para uma demonstração das possibilidades que o pacote oferece utilize o comando:

```
demo(googleVis)
```

Neste caso você vai apertando a tecla *enter* e a cada vez será gerado um gráfico no seu navegador padrão.

5.2 Os gráficos de movimento (*Motion Chart*) do Google

Neste capítulo vamos mostrar como é possível criar um gráfico de movimentos.

O primeiro passo é possuir uma planilha com a seguinte arquitetura:

Coluna 1: variável quantitativa temporal: em geral ano

Coluna 2: variável qualitativa

Colunas 3 e 4: variável numérica

Esta é a estrutura mínima da sua planilha de dados, podendo ser acrescentado colunas de qualquer natureza após a quarta coluna. Note que a ordem das quatro primeiras colunas quanto à natureza da variável deve ser respeitada para que tudo funcione corretamente.

Vamos para um exemplo prático para o Comando *gvisMotionChart*

Estamos interessados em comparar o comportamento do índice de aprovação de três cursos diferentes (A, B e C) em duas instituições (UFF e UERJ) no período de 2010 a 2013.

Nossa planilha de dados pode ser vista na tabela 5.1:

Tabela 5.1: Dados sobre o índice de aprovação de alunos

Ano	curso	UFF	UERJ
2010	A	20	50
2011	A	30	40
2012	A	35	46
2013	A	44	43
2010	B	9	30
2011	B	20	28

2012	B	13	33
2013	B	30	23
2010	C	30	15
2011	C	10	19
2012	C	18	21
2013	C	4	24

Fonte: Dados Fictícios

Suponha que a tabela esteja armazenada no arquivo *BD.csv*. É importante que a sequência do tempo da coluna 1 se repita igual para cada categoria da coluna 2. Desse modo, o banco de dados deverá ter número de linhas igual ao comprimento do período multiplicado pelo número de categorias. No nosso exemplo será $4 \times 3 = 12$.

Script para gerar o gráfico de movimento

```
dados <- read.csv2("C:\\...\\BD.csv")

M=gvisMotionChart(dados, idvar="curso", timevar="Ano", options=list(height=350,
width=400), )

plot(M)
```

#Para detalhes sobre a função utilize:

?gvisMotionChart

O resultado será o gráfico produzido em *html* (Figura 5.1). Diversos recursos podem ser explorados neste gráfico: colocá-lo em movimento, observando o que acontece em relação ao tempo e também o que acontece com a relação entre o índice de aprovação da UFF e UERJ para cada curso, tudo ao mesmo tempo! Podemos ainda selecionar a cor das bolas de acordo com uma das duas variáveis UFF ou UERJ. Na figura 5.1 podemos observar a coloração aplicada quando escolhemos a variável UFF. Neste caso, sendo UFF uma variável quantitativa, cada valor corresponderá a uma cor de acordo com o termômetro de cores, quanto maior o valor mais quente é a cor.

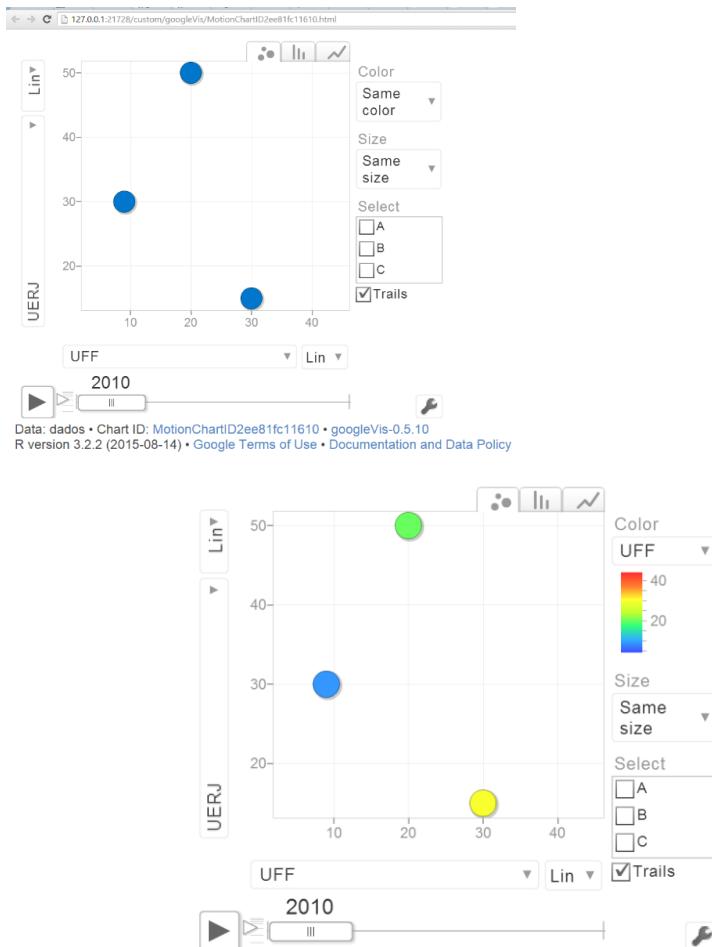


Figura 5-1: Gráficos de Movimento – Bolhas

É possível visualizar três tipos de gráfico disponível: bolhas, barras e linhas, além das funcionalidades disponíveis no menu lateral como *Color*, *Size* e *Select*. *Size* modifica o tamanho da bolha de acordo com a variável selecionada e *Select* possibilita acompanhar o comportamento de uma ou mais categorias que foram selecionadas. Isso gera uma grande quantidade de diferentes gráficos na mesma janela, possibilitando ao usuário fazer diversas explorações, comparações e análises.

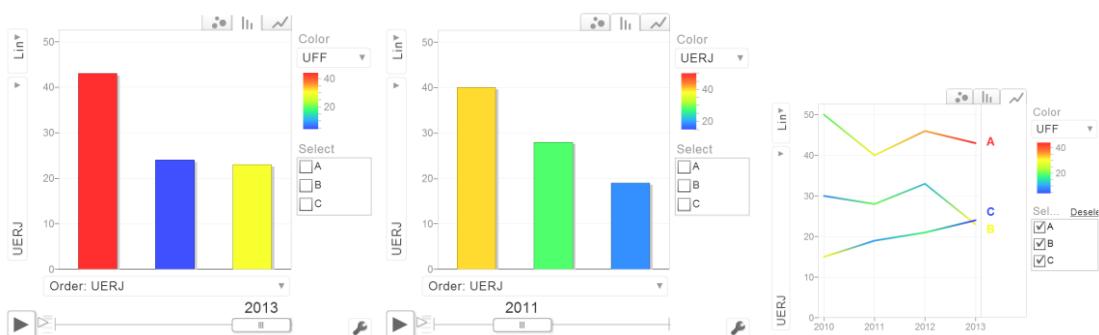


Figura 5-2 Gráficos de Movimento: Barras e Linhas

Além destas opções é possível utilizar um *script* para realizar edições online, abrindo uma grande variedade de gráficos que podem ser modificados instantaneamente. Para isto usamos o argumento *Edit me!*

Script para editar o layout do gráfico instantaneamente

```
M=gvisMotionChart(dados, idvar="curso", timevar="Ano",
options=list(gvis.editor="Edit me!", height=350, width=400))
```

Plot(M)

Neste exemplo, além das possibilidades descritas acima, abrirá um editor de gráficos ao lado direito da janela, cujas opções podem ser vistas na Figura 5.3.

Editor de gráficos



Figura 5-3: Editor de gráficos do google usando o argumento “Edit me!”

Entretanto, apesar das múltiplas opções o analista deve ficar atento pois nem todo gráfico é aplicável para o banco de dados que dispõe. Neste exemplo, a melhor alternativa é o gráfico de bolhas.

Você já deve ter percebido que todos esses recursos não terão tanto impacto nas formas tradicionais de exposição como documentos impressos, documentos em *pdf* ou apresentações em slide já que estes gráficos permitem o movimento e são gerados em código *html*, foram projetados para serem exibidos na *web* ou dispositivo que permita o movimento.

5.3 Gráficos de Linha do Google

Usaremos o comando `gvisLineChart`. De forma muita rápida podemos gerar um gráfico de linhas, comparando duas ou mais séries de valores correspondentes à categorias.

A arquitetura para este tipo de gráfico é um banco de dados com 3 colunas, uma variável qualitativa e duas ou mais quantitativas para o caso de comparações, veja a tabela de dados 5.2:

Tabela 5.2: Dados sobre o número de homens (H) e mulheres (M) em cada curso

curso	H	M
A	20	50
B	30	40
C	35	46
D	44	43
E	9	30

Fonte: Dados Fictícios

Script para gerar o gráfico de Linha do pacote googleVis

```
dados <- read.csv2("C:\...\BD.csv")  
L=gvisLineChart(dados, xvar="curso", yvar=c("H", "M"),  
options=list(gvis.editor="Edit me!", height=350, width=400) )  
plot(L)
```

Com o *script* acima você gerará um gráfico simples de linha para as variáveis H e M. Note que o argumento *xvar* deve ser sempre uma variável qualitativa, não pode haver espaço entre as aspas e o nome do vetor (“curso”) e o *yvar* deve ser vetor de variável quantitativa.

Ao explorar os recursos gerado pelo comando `plot(L)`, você terá a opção de gerar gráficos de linhas e barras de diversas formas, acrescentar títulos, etc. Para se ter uma ideia, selecionamos a gráfico de barras com linhas, veja figura 5.4

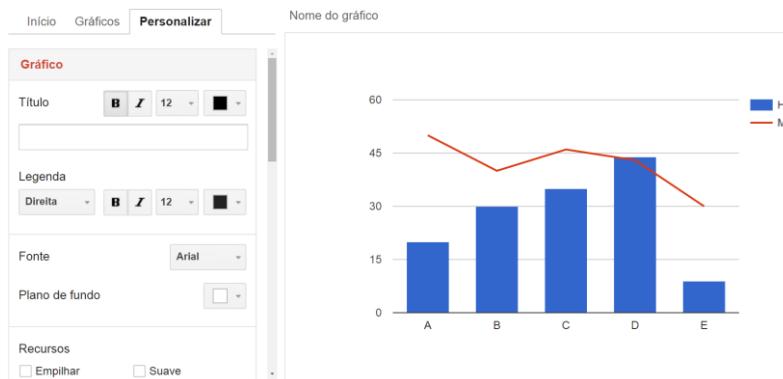


Figura 5-4: Editor de gráficos do google usando o argumento “Edit me!”

Outro exemplo, agora personalizando título, posicionamento da legenda, rótulo dos eixos, etc. Explore os recursos, pois são muitos!

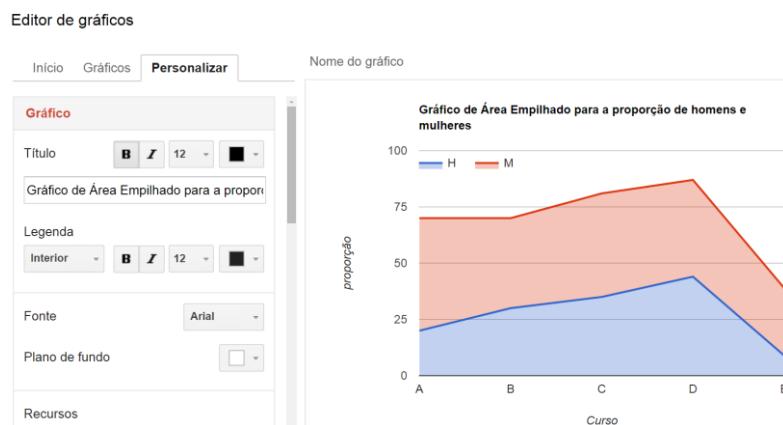


Figura 5-5: Editor de gráficos do google Personalização

5.3.1 Hora de Praticar

1. Lembra-se do script do capítulo 2 usando pacote básico do R para duas séries de valores? Veja:

Script para plotar duas séries temporais em gráfico de linha com legenda

```
ano<-2001:2009
tri1<-c(72.8,66.2,69.2,65.9,62.4,67.8,61.3,68.5,70.4)
tri2<-c(60.6,53.7,55.3,56.7,56.4,57.8,57.5,59.8,63.3)
plot(ano, tri1, type="b", main="Taxa de ocupação por trimestre dos hotéis - Município do Rio de Janeiro", xlab="ano", ylab="Taxa de ocupação %", col="blue",
ylim=c(50,80))
lines(ano, tri2,col="red", type="b")
legend(2007, 80, c("1o.trim","2o.trim"), col =c("blue","red"), pch=rep(20,2))
```

Crie um data.frame para armazenar os 3 vetores de dados: ano, tri1 e tri2 Utilize estes dados para fazer o gráfico no Google.

2. Considere os dados da Tabela sobre a freqüência para um grupo de fumantes de cigarro e um grupo de não-fumantes em diversos níveis séricos de cotinina. Proponha um gráfico de linhas do Google

<i>Nível de Cotinina (ng/ml)</i>	<i>Fumantes</i>	<i>Não-fumantes</i>
0 –13	78	3300
14 – 49	133	72
50 – 99	142	23
100 – 149	206	15
150 – 199	197	7
200 – 249	220	8
250 – 299	151	9
300 +	412	11

3. Proponha um gráfico de linha do google para os dados abaixo.

Área do conhecimento	Instituição A	Instituição B
Saúde	1751	2528
Exatas	2186	2132
Humanas	947	1843
Comunicação	29	280

5.4 Gráficos de Setores do Google

Vamos refazer os gráficos da seção 2.5. Note que o *script* é muito mais simples e depois de gerado o arquivo *html*, muitas opções de edição são possíveis. Duas possibilidades são exibidas na figura 5-6.

Script para gráfico de setores no Google

```
frota<-c(151882, 19627, 8058,1895, 3201, 1969,1268)  
nomes=c("automoveis","motocicletas","caminhonetas","motonetas","onibus","caminhoes", "outros")  
dados=data.frame(nomes, frota)  
P=gvisPieChart(dados, options=list(gvis. editor="Edit me!", height=350, width=400))  
plot(P)
```

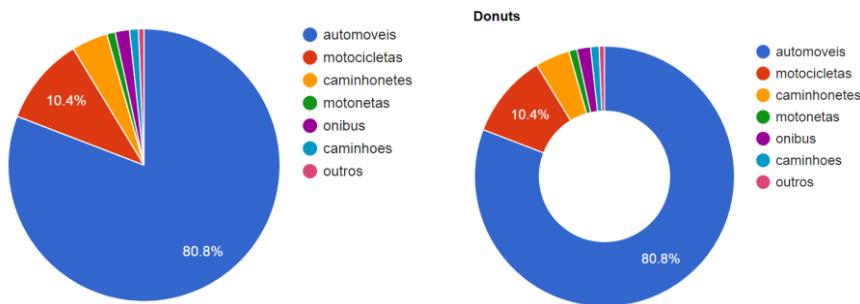


Figura 5-6: Possibilidades para o gráfico de setores do google

5.4.1 Hora de Praticar

Refaça a Seção 2.5.1, usando os recursos do *googleVis* para propor representações web.

5.5 Gauge

Este tipo de gráfico é útil para monitorar processos, as variáveis são visualizadas por indicadores na forma de ponteiro, como se fosse o velocímetro de um carro.

Vamos produzir *gauges* para representar o percentual de homens em cinco instituições diferentes A, B, C, D e E cujos valores são respectivamente 20, 30, 35, 44 e 9.

Definiremos 3 faixas de cores, verde com percentual de 0 a 30; amarelo com percentual de 30 a 70 e vermelho com percentual de 70 a 100.

Script para Gauge

Col1=LETTERS[1:5]; Col2=c(20,30,35,44,9)

```
Gauge <- gvisGauge(data.frame(Col1,Col2), options=list(min=0, max=100,  
greenFrom=0, greenTo=30, yellowFrom=30, yellowTo=70, redFrom=70, redTo=100))  
plot(Gauge)
```



Figura 5-7: Gauges do google

Podemos interpretar estes gráficos observando seus ponteiros. No caso em tela, vemos que a instituição A e E possuem menos de 30% de homens pois os ponteiros estão na faixa verde cujo percentual varia de 0 a 30.

Aqui também é possível incluir o comando *Edit me!* que torna possível mudar os intervalos das cores e o intervalo de valores.

Estes gráficos se aplicam em situações onde se deseja observar um indicador numérico que será representado pelo ponteiro do relógio.

5.5.1 Hora de Praticar

- 1- Uma pesquisa de satisfação sobre os itens, sabor e aroma de duas marcas de café geraram as seguintes notas em uma escala de 0 a 10:

Marca	Aroma	Sabor
A	7	5
B	8.5	8.3

Elabore um *gauge* para indicar estas notas para cada marca.

- 2- Três corredores que disputaram uma maratona obtiveram o seguinte desempenho em velocidade média do percurso (km/h): 18.5; 19.4 e 21.3. Elabore um *gauge* para representar a velocidade de cada corredor.
- 3- Considere a previsão de temperatura em graus Celsius para certa localidade nos próximos 5 dias. Projete um *gauge* que indique três faixas de valores: 0 a 15; 15 a 30 e 30 a 50, supondo que os valores previstos sejam 22; 31; 25; 19; 18; 15

5.6 Gráficos Geográficos do Google

Podemos também produzir visualização de dados em mapa geográfico. Vamos mostrar como deve ser a arquitetura do arquivo de dados dos países para produzir uma visualização no mapa *mundi*.

O primeiro passo é saber o nome canônico do país. Para isso consulte o Apêndice A.

Coluna 1: Nome do país

Coluna 2: variável quantitativa relacionada à localidade da coluna 1

Exemplo 3: Considere os dados da expectativa de vida de oito países.

Tabela 5.3: Dados sobre Esperança de vida ao nascer

Pais	E.Vida
Sierra Leone	46
India	66
Tonga	71
Brasil	74
China	75
Portugal	81
France	82
Japan	84

Fonte: World Health Organization (2013)

Supondo que estes dados estejam em planilha *BD.csv*, ao rodar o *script* abaixo, o resultado será semelhante ao da figura 5.8.

Script para gerar o Geográfico do pacote googleVis

```
dados <- read.csv2("C:\\...\\BD.csv")  
G = gvisGeoChart(dados, locationvar='Pais', colorvar='E.Vida')  
plot(G)
```

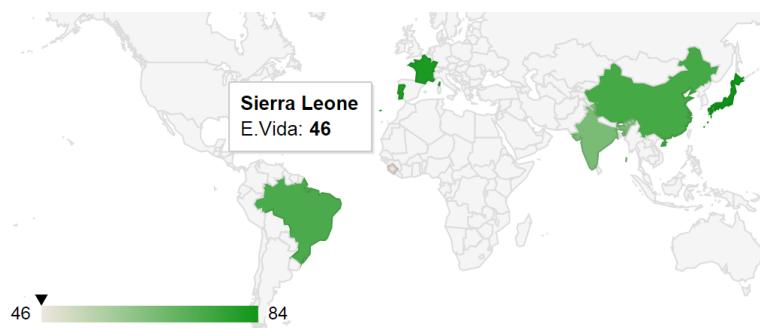


Figura 5-8: Geográfico Padrão do Google

Usando o recurso da função *Edit me!*, é possível produzir com o mesmo comando e algumas manipulações simples no menu disponível, gerando um gráfico como o da figura 5.9, com o comando abaixo:

```
G= gvisGeoChart(dados, locationvar='Pais', colorvar='E.Vida',  
options=list(gvis.editor="Edit me!") )
```

O menu permite que o usuário escolha as cores para a legenda, a cor do plano de fundo, o título do gráfico, além de escolher a região do mapa que deseja visualizar, cujas opções são:

Ásia; África; Europa; América do Norte; América do Sul; Oceania; Estados Unidos e Internacional que engloba todas as opções anteriores. Nas figuras 5.8 e 5.9 o que se vê é a opção Internacional. Já o gráfico 5.10 refere-se a região da Ásia.

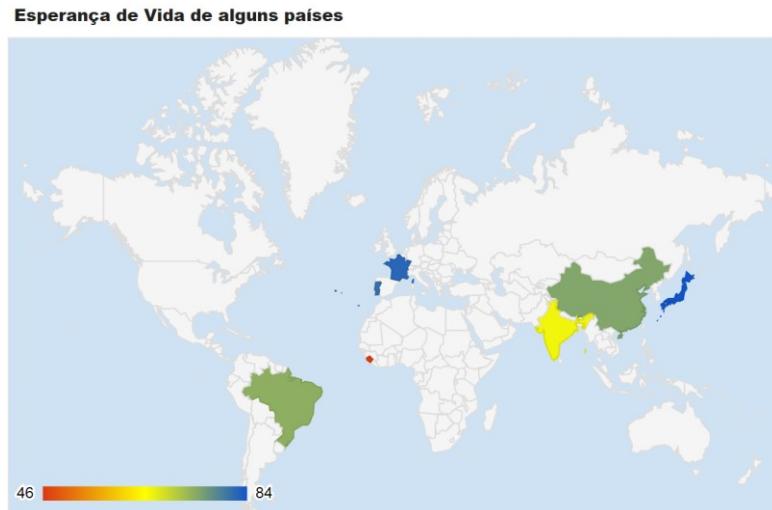


Figura 5-9: Geográfico editado no Google

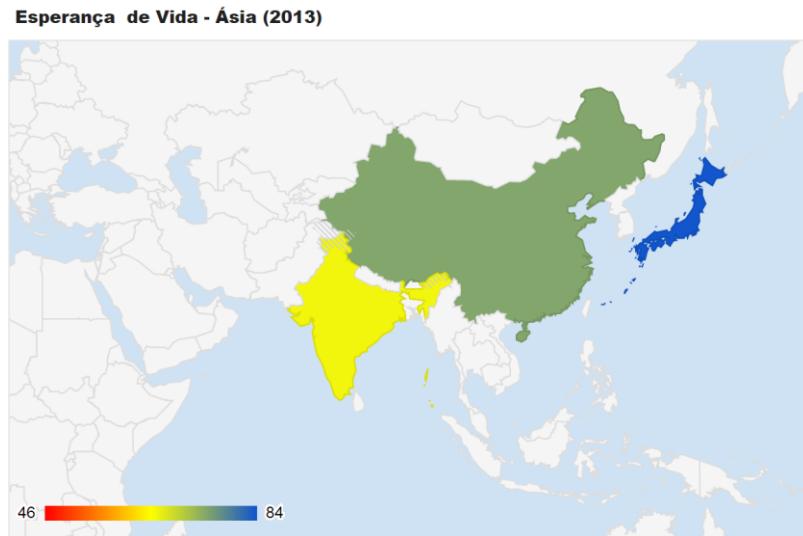


Figura 5-10: Geográfico editado no Google, região Ásia

Pode ser útil publicar a tabela de dados que gerou o geográfico. Neste caso use usaremos adicionalmente a função *gvisTable* para definir a tabela e sua dimensão na tela e a função *gvisMerge* para mesclar o geográfico e a tabela na mesma tela. O argumento horizontal=TRUE se omitido posicionará a tabela abaixo do gráfico e não ao lado como pode ser visto na figura 5.11.

Script para Geográfico e Tabela de dados

```
G= gvisGeoChart(dados, locationvar='Pais', colorvar='E.Vida', options =
list(width=400, height=400))

T= gvisTable(dados, options = list(width=150, height=200))

GT= gvisMerge(G, T, horizontal=TRUE)

plot(GT)
```

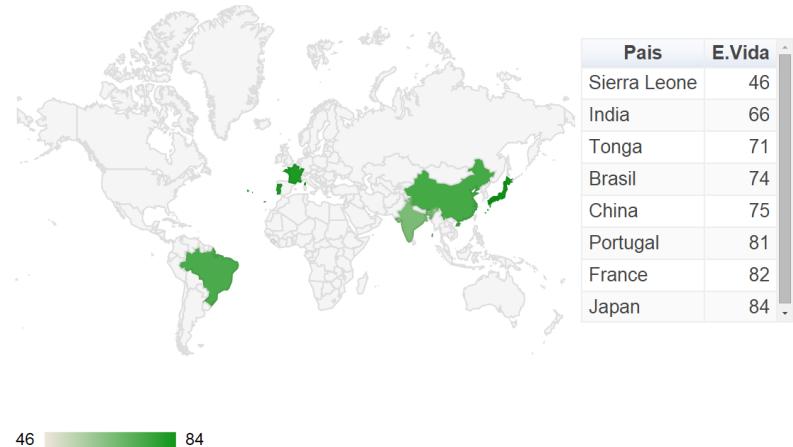


Figura 5-11: Geográfico e Tabela de dados na mesma janela

Podemos ainda, imprimir marcas no mapa para representar uma variável georeferenciada e especificar a sua coloração.

```
G= gvisGeoMap(dados, locationvar='Pais', numvar= 'E.Vida', hovervar= 'Pais',
options = list(dataMode= 'markers', colors='[0xFF8747, 0xFFB581, 0xc06000]'))
```

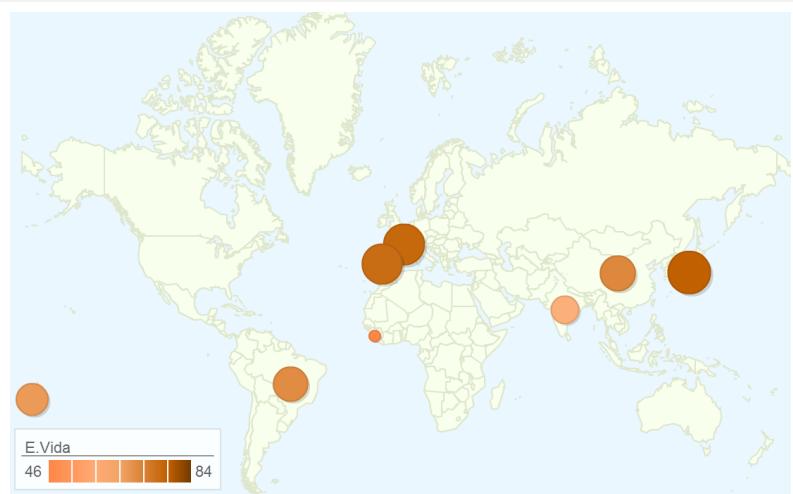


Figura 5-12: Geográfico com marcas para Esperança de Vida de alguns países.

Esperamos com esse capítulo ter mostrado a potencialidade de uso deste pacote, mostrando como o R pode revolucionar de forma acessível a maneira como podemos representar nossos dados. Agora é praticar e explorar os recursos possíveis.

5.6.1 Hora de Praticar

- 1- Produza um banco de dados com a população dos países da América do Sul. Utilize o apêndice A para obter os nomes canônicos. Produza um geográfico com edições possíveis.
- 2- Produza um geográfico mundial com o IDH dos países.
- 3- Produza um geográfico da América do Sul com o PIB dos países.

5.7 Referência Bibliográfica do capítulo 5

Gesmann, M; Castillo, D. **Using the Google Visualisation API with R.** The R Journal, 3(2):40-44, December 2011.

Gesmann, G., Castillo, D., Cheng, J., **Package googleVis.** Disponível em <https://cran.r-project.org/web/packages/googleVis/googleVis.pdf>. Acesso em 13/12/2015.

Hans Rosling. **TED Talk: Hans Rosling shows the best stats you've ever seen.** Disponível em http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen, , 2006. Acesso em 08/12/2015.

6 Dispositivos Gráficos e Cores no R

Ariel Levy

Os gráficos em geral não se destinam apenas para verificação na tela do computador, mas para compor apresentações, artigos enfim documentos. Vários formatos podem ser obtidos com o R, tais como: EPS, PNG, SVG, TIFF, WMF, etc.

Quando criamos um gráfico precisamos considerar qual será o dispositivo em que ele será mostrado. A lista de possíveis dispositivos pode ser obtida no CRAN (<https://stat.ethz.ch/R-manual/R-devel/library/grDevices/html/Devices.html>).

Enquanto exploramos dados de forma interativa a tela parece ser o dispositivo mais indicado e o formato *pdf* é o padrão para os demais casos.

Assim, quando executamos um comando de gráfico como *plot*, no pacote *basic*, *xyplot*, no *lattice* ou *qplot*, no *ggplot2* estaremos por padrão enviando um gráfico para a tela que conforme a plataforma tem um nome: no Windows será *Windows()*, no Mac denomina-se *quartz()* e no Linux, será *x11()*.

Haverá momentos em que preferiremos arquivar os gráficos para uso futuro em outros formatos de documentos, para os quais usamos os dispositivos.

Estes formatos são classificados como Vetorizados e Bitmap. Entender suas diferenças pode ajudar na futura adequação do objeto a finalidade pretendida.

Os formatos vetorizados permitem melhor redimensionamento e se adequam melhor a linhas e formas. São exemplos de formatos vetorizados o PDF, SVG, Win.metafile, este último disponível apenas no Windows.

PDF- portable document format - São portáveis, redimensionáveis, mas não muito eficientes quando uma imagem tem muitos pontos ou detalhes, os arquivos ficam grandes.

SVG- scalable vector graphics – só pelo nome já vemos que são adequados ao redimensionamento, normalmente muito utilizados em gráficos animados e interativos.

Também o formato POSTSCRIPT é vetorizado, entretanto o Windows não apresenta um visualizador nativo para este tipo de arquivo.

Os Bitmaps definem a exposição e cor para cada pixel ou bit, são exemplos deste tipo JPEG, PNG, GIF, TIFF e BMP. Este tipo de arquivo utiliza um método de especificação que

melhor reproduz áreas com cores sólidas requerendo menos informação do que o tipo vetorizado, mas perde resolução mais facilmente quando redimensionados.

Quando direcionamos um gráfico para outro dispositivo ele não será mostrado na tela então o cuidado na execução deverá ser maior.

O formato PNG e JPEG podem ser facilmente convertidos para muitos outros formatos Bitmap. Este formato permite menores perdas no redimensionamento e trabalha bem com diagramas e blocos de cor.

```
# apenas para formar o exemplo crie a sequência  
x<-seq(1,100, 0.1)  
  
meu_grafico<-sin(x)  
  
# abrindo o dispositivo png  
  
png("meu_grafico", width=480, height=480)  
  
# feche o dispositivo  
  
dev.off()
```

Veja que o código criará no diretório de trabalho o arquivo “meu_grafico.png”, e o mesmo não será mostrado na tela. Recomenda-se verificar o arquivo gerado abrindo-o com o programa correspondente. Neste caso o leitor verificará por exemplo que faltam legendas nos eixos e um título, que já foi objeto de capítulos anteriores. A título de exercício modifique o código aprimorando “meu_grafico.png”.

O mesmo procedimento pode ser realizado para os demais dispositivos.

O pacote *R.devices* permite controlar diversas opções de forma uniforme para todos os tipos de tipos de dispositivos. Produzindo saídas em diversos formatos com alterações mínimas no código. Sendo ideal para processamento em “batch”, automatizado, porque nunca se esquecerá um dispositivo aberto, minimizando erros ou imagens incompletas. Um hábito que o leitor deve adquirir é ler os *vignettes* dos pacotes. As informações em geral são suficientes para um uso adequado dos mesmos. Os *vignettes* são disponibilizados e livres para consulta a partir do CRAN. No console do R é possível obter vignettes em pacotes instalados ou em uso:

```
vignette()          #      mostra      os      vignettes      disponíveis  
vignette("R.devices") # mostra um vignette específico
```

No índice do CRAN relativo ao pacote será possível baixar o manual no formato pdf dos pacotes. Sua leitura trará ao leitor a compreensão dos detalhes a partir da visualização e recomendada repetição no console dos exemplos. Aprende-se melhor fazendo e verificando os resultados, embora isso vá de encontro ao in=mediatismo da tarefa almejada.

6.1 Utilizando Múltiplos Dispositivos

Embora seja possível abrir mais de um dispositivo simultaneamente estaremos criando o gráfico em apenas um por vez. Pode-se verificar qual o dispositivo ativo com a função `dev.cur()` que retorna um número inteiro maior ou igual a 2. Também podemos alterar qual o dispositivo ativo com a função `dev.set(<inteiro>)` onde o <inteiro> corresponde ao número associado ao dispositivo gráfico desejado.

Vejamos um exemplo: Suponha que você deseje escolher quais dos três gráficos gostaria de adicionar a sua apresentação.

```
# criar 3 gráficos em janelas diferentes  
dev.new() # ou windows() ou quartz() ou x11() conforme a plataforma  
dev.1<- as.integer(dev.cur())  
dev.new()  
dev.2<- as.integer(dev.cur())  
dev.new()  
dev.3<- as.integer(dev.cur())  
x<-seq(1,100, 0.1)
```

```
# trocando para o dispositivo 1  
dev.set(dev.1)  
plot(x, sin(x), "1")  
# trocando para o dispositivo 3  
dev.set(dev.3)
```

```

plot(x, cos(x), "1")

# voltando para o dispositivo 1 e alterando o gráfico

dev.set(dev.1)

plot(x, sin(x), "1", col="red")

```

Sempre deve-se recorrer ao comando `dev.cur()` ao criar novos dispositivos pois não podemos afirmar que serão criados na ordem (1,2,3) já que podem haver outros dispositivos abertos.

6.2 Gráficos e suas Cores no R

As cores padrões utilizadas nos pacotes de gráficos do R não favorecem a aparência ou os aspectos decisórios. Mas muitos desenvolvimentos foram realizados neste sentido. Um bom exemplo são as funções `colorRamp` e `colorRampPalette` presentes no pacote `grDevices`. Elas permitem a interpolação ou mistura de cores. Outra função que merece registro é a função `colors()` que lista todas as 657 cores disponíveis por nome para utilização em gráficos.

Obter um bom esquema de cores pode ser desafiante ao buscar ressaltar os aspectos desejados numa imagem. O ditado popular já coloca a importância da imagem: “uma imagem vale mais que 1000 palavras”. Aspectos como transparência tipo do símbolo e separação categórica já foram abordados antes, mas entender como o R lida com as cores lhe dará liberdade em suas criações.

O R utiliza uma representação hexadecimal, base 16, para descrever uma cor.

O sistema RGB (vermelho, verde, azul) é representado por dois caracteres cada (#rr,gg,bb). Cada qual podendo variar de 0 a F, lembre-se base hexadecimal.

Assim, “00” poderá ser interpretado como zero e “FF” como 100%. Por exemplo: #FFFFFF é o branco, #000000 o preto, #0000FF azul, #00FF00 verde.

Quando desejar-se incluir a transparência na cor adiciona-se mais dois caracteres ao final (#rrggbbaa).

O R pode traduzir o modelo RGB para hexadecimal. Suas escalas variam de 0-1 mas outra usual é a de 0-255, que pode ser obtida usando `maxColorValue=255`. O `alpha` é o argumento utilizado para transparência, com a mesma escala variando de 0 a 1, utilizamos anteriormente no gráfico de listas no capítulo do *Lattice*.

Se o leitor não quer perder muito tempo, ou iniciar-se na teoria das cores sugerimos que utilize uma paleta de cores pré-definido no pacote *grDevices*, tais como: *cm.colors*, *topo.colors*, *terrain.colors*, *heat.colors* *rainbow* etc.

O pacote *RColorBrewer* apresenta paletas categorizadas para utilização conforme objetivos a ressaltar: *sequencial* - destinada a dados numéricos relacionados a uma tendência, *diverging* – para dados que se afastam de uma média ou tendência, ou *categorical* – para separação condicional a uma variável categórica. Uma visualização completa destas paletas pode ser obtida com o pacote instalado e carregado através do comando:

```
display.brewer.all(5)
```

Por certo isto não esgota o tema das possibilidades gráficas do R com seus dispositivos e cores, mas dá ao leitor um marco inicial para sua exploração e aprimoramento.

6.3 Referência Bibliográfica do capítulo 6

BENGSSON, Henrik **Package R.devices.** Disponível em: <https://cran.r-project.org/web/packages/R.devices/R.devices.pdf> . Acesso em 10/12/2015/

BRYAN, Jennifer **Using Colors in R.** Disponível em:

http://www.stat.ubc.ca/~jenny/STAT545A/block14_colors.html . Acesso em: 12/12/2015

GLYNN, Earl F. **Pdf of Chart R Colors.** Disponível em: <http://research.stowers-institute.org/efg/R/Color/Chart/> . Acesso em 11/12/2015

R documentation, **Control Multiple devices.** Disponivel em:

<http://astrostatistics.psu.edu/datasets/R/html/grDevices/html/dev.html> . Acesso em 10/12/2015

R documentation, **Color Palettes**, Disponível em: <https://stat.ethz.ch/R-manual/R-devel/library/grDevices/html/palettes.html> . Acesso em 14/12/2015.

8 Apêndice A

The ISO-3166-1 alpha-2 country code that is associated with the target.

Fonte: <https://developers.google.com/adwords/api/docs/appendix/geotargeting>

Critério ID	Nome	Nome Canônico	Código do País	Tipo
1	2004	Afghanistan	AF	Country
2	2008	Albania	AL	Country
3	2010	Antarctica	AQ	Country
4	2012	Algeria	DZ	Country
5	2016	American Samoa	AS	Country
6	2020	Andorra	AD	Country
7	2024	Angola	AO	Country
8	2028	Antigua and Barbuda	AG	Country
9	2031	Azerbaijan	AZ	Country
10	2032	Argentina	AR	Country
11	2036	Australia	AU	Country
12	2040	Austria	AT	Country
13	2044	The Bahamas	BS	Country
14	2048	Bahrain	BH	Country
15	2050	Bangladesh	BD	Country
16	2051	Armenia	AM	Country
17	2052	Barbados	BB	Country
18	2056	Belgium	BE	Country
19	2064	Bhutan	BT	Country
20	2068	Bolivia	BO	Country
	Bosnia and Herzegovina	Bosnia and Herzegovina	BA	Country
21	2070	Botswana	BW	Country
22	2072	Brazil	BR	Country
23	2076	Belize	BZ	Country
24	2084	Solomon Islands	SB	Country
25	2090	Brunei	BN	Country
26	2096	Bulgaria	BG	Country
27	2100	Burundi	BI	Country
28	2108	Belarus	BY	Country
29	2116	Cambodia	KH	Country
30	2120	Cameroon	CM	Country
31	2124	Canada	CA	Country
32	2132	Cape Verde	CV	Country
34	2140	Central African Republic	CF	Country
35	2144	Sri Lanka	LK	Country
36	2148	Chad	TD	Country
37	2152	Chile	CL	Country
38	2156	China	CN	Country
39	2162	Christmas Island	CX	Country
40	2166	Cocos (Keeling) Islands	CC	Country
41	2170	Colombia	CO	Country
42	2174	Comoros	KM	Country
43	2178	Congo	CG	Country
44	2180	Democratic Republic of the Congo	CD	Country
45	2184	Cook Islands	CK	Country
46	2188	Costa Rica	CR	Country
47	2191	Croatia	HR	Country
48	2196	Cyprus	CY	Country
49	2203	Czech Republic	CZ	Country

50	2204	Benin	Benin	BJ	Country
51	2208	Denmark	Denmark	DK	Country
52	2212	Dominica	Dominica	DM	Country
53	2214	Dominican Republic	Dominican Republic	DO	Country
54	2218	Ecuador	Ecuador	EC	Country
55	2222	El Salvador	El Salvador	SV	Country
56	2226	Equatorial Guinea	Equatorial Guinea	GQ	Country
57	2231	Ethiopia	Ethiopia	ET	Country
58	2232	Eritrea	Eritrea	ER	Country
59	2233	Estonia	Estonia	EE	Country
60	2239	South Georgia and the South Sandwich Islands	South Georgia and the South Sandwich Islands	GS	Country
61	2242	Fiji	Fiji	FJ	Country
62	2246	Finland	Finland	FI	Country
63	2250	France	France	FR	Country
64	2258	French Polynesia	French Polynesia	PF	Country
65	2260	French Southern and Antarctic Lands	French Southern and Antarctic Lands	TF	Country
66	2262	Djibouti	Djibouti	DJ	Country
67	2266	Gabon	Gabon	GA	Country
68	2268	Georgia	Georgia	GE	Country
69	2270	The Gambia	The Gambia	GM	Country
70	2276	Germany	Germany	DE	Country
71	2288	Ghana	Ghana	GH	Country
72	2296	Kiribati	Kiribati	KI	Country
73	2300	Greece	Greece	GR	Country
74	2308	Grenada	Grenada	GD	Country
75	2316	Guam	Guam	GU	Country
76	2320	Guatemala	Guatemala	GT	Country
77	2324	Guinea	Guinea	GN	Country
78	2328	Guyana	Guyana	GY	Country
79	2332	Haiti	Haiti	HT	Country
80	2334	Heard Island and McDonald Islands	Heard Island and McDonald Islands	HM	Country
81	2336	Vatican City	Vatican City	VA	Country
82	2340	Honduras	Honduras	HN	Country
83	2348	Hungary	Hungary	HU	Country
84	2352	Iceland	Iceland	IS	Country
85	2356	India	India	IN	Country
86	2360	Indonesia	Indonesia	ID	Country
87	2368	Iraq	Iraq	IQ	Country
88	2372	Ireland	Ireland	IE	Country
89	2376	Israel	Israel	IL	Country
90	2380	Italy	Italy	IT	Country
91	2384	Cote d'Ivoire	Cote d'Ivoire	CI	Country
92	2388	Jamaica	Jamaica	JM	Country
93	2392	Japan	Japan	JP	Country
94	2398	Kazakhstan	Kazakhstan	KZ	Country
95	2400	Jordan	Jordan	JO	Country
96	2404	Kenya	Kenya	KE	Country
97	2410	South Korea	South Korea	KR	Country
98	2414	Kuwait	Kuwait	KW	Country
99	2417	Kyrgyzstan	Kyrgyzstan	KG	Country
100	2418	Laos	Laos	LA	Country
101	2422	Lebanon	Lebanon	LB	Country
102	2426	Lesotho	Lesotho	LS	Country
103	2428	Latvia	Latvia	LV	Country
104	2430	Liberia	Liberia	LR	Country
105	2434	Libya	Libya	LY	Country
106	2438	Liechtenstein	Liechtenstein	LI	Country
107	2440	Lithuania	Lithuania	LT	Country
108	2442	Luxembourg	Luxembourg	LU	Country
109	2450	Madagascar	Madagascar	MG	Country

110	2454	Malawi	Malawi	MW	Country
111	2458	Malaysia	Malaysia	MY	Country
112	2462	Maldives	Maldives	MV	Country
113	2466	Mali	Mali	ML	Country
114	2470	Malta	Malta	MT	Country
115	2478	Mauritania	Mauritania	MR	Country
116	2480	Mauritius	Mauritius	MU	Country
117	2484	Mexico	Mexico	MX	Country
118	2492	Monaco	Monaco	MC	Country
119	2496	Mongolia	Mongolia	MN	Country
120	2498	Moldova	Moldova	MD	Country
121	2499	Montenegro	Montenegro	ME	Country
122	2504	Morocco	Morocco	MA	Country
123	2508	Mozambique	Mozambique	MZ	Country
124	2512	Oman	Oman	OM	Country
125	2516	Namibia	Namibia	NA	Country
126	2520	Nauru	Nauru	NR	Country
127	2524	Nepal	Nepal	NP	Country
128	2528	Netherlands	Netherlands	NL	Country
129	2540	New Caledonia	New Caledonia	NC	Country
130	2548	Vanuatu	Vanuatu	VU	Country
131	2554	New Zealand	New Zealand	NZ	Country
132	2558	Nicaragua	Nicaragua	NI	Country
133	2562	Niger	Niger	NE	Country
134	2566	Nigeria	Nigeria	NG	Country
135	2570	Niue	Niue	NU	Country
136	2574	Norfolk Island	Norfolk Island	NF	Country
137	2578	Norway	Norway	NO	Country
138	2580	Northern Mariana Islands	Northern Mariana Islands	MP	Country
139	2581	United States Minor Outlying Islands	United States Minor Outlying Islands	UM	Country
140	2583	Micronesia	Micronesia	FM	Country
141	2584	Marshall Islands	Marshall Islands	MH	Country
142	2585	Palau	Palau	PW	Country
143	2586	Pakistan	Pakistan	PK	Country
144	2591	Panama	Panama	PA	Country
145	2598	Papua New Guinea	Papua New Guinea	PG	Country
146	2600	Paraguay	Paraguay	PY	Country
147	2604	Peru	Peru	PE	Country
148	2608	Philippines	Philippines	PH	Country
149	2612	Pitcairn Islands	Pitcairn Islands	PN	Country
150	2616	Poland	Poland	PL	Country
151	2620	Portugal	Portugal	PT	Country
152	2624	Guinea-Bissau	Guinea-Bissau	GW	Country
153	2626	East Timor	East Timor	TL	Country
154	2634	Qatar	Qatar	QA	Country
155	2642	Romania	Romania	RO	Country
156	2643	Russia	Russia	RU	Country
157	2646	Rwanda	Rwanda	RW	Country
158	2654	Saint Helena	Saint Helena	SH	Country
159	2659	Saint Kitts and Nevis	Saint Kitts and Nevis	KN	Country
160	2662	Saint Lucia	Saint Lucia	LC	Country
161	2666	Saint Pierre and Miquelon	Saint Pierre and Miquelon	PM	Country
162	2670	Saint Vincent and the Grenadines	Saint Vincent and the Grenadines	VC	Country
163	2674	San Marino	San Marino	SM	Country
164	2678	Sao Tome and Principe	Sao Tome and Principe	ST	Country
165	2682	Saudi Arabia	Saudi Arabia	SA	Country
166	2686	Senegal	Senegal	SN	Country
167	2688	Serbia	Serbia	RS	Country
168	2690	Seychelles	Seychelles	SC	Country

169	2694	Sierra Leone	Sierra Leone	SL	Country
170	2702	Singapore	Singapore	SG	Country
171	2703	Slovakia	Slovakia	SK	Country
172	2704	Vietnam	Vietnam	VN	Country
173	2705	Slovenia	Slovenia	SI	Country
174	2706	Somalia	Somalia	SO	Country
175	2710	South Africa	South Africa	ZA	Country
176	2716	Zimbabwe	Zimbabwe	ZW	Country
177	2724	Spain	Spain	ES	Country
178	2740	Suriname	Suriname	SR	Country
179	2748	Swaziland	Swaziland	SZ	Country
180	2752	Sweden	Sweden	SE	Country
181	2756	Switzerland	Switzerland	CH	Country
182	2762	Tajikistan	Tajikistan	TJ	Country
183	2764	Thailand	Thailand	TH	Country
184	2768	Togo	Togo	TG	Country
185	2772	Tokelau	Tokelau	TK	Country
186	2776	Tonga	Tonga	TO	Country
187	2780	Trinidad and Tobago	Trinidad and Tobago	TT	Country
188	2784	United Arab Emirates	United Arab Emirates	AE	Country
189	2788	Tunisia	Tunisia	TN	Country
190	2792	Turkey	Turkey	TR	Country
191	2795	Turkmenistan	Turkmenistan	TM	Country
192	2798	Tuvalu	Tuvalu	TV	Country
193	2800	Uganda	Uganda	UG	Country
194	2804	Ukraine	Ukraine	UA	Country
195	2807	Macedonia (FYROM)	Macedonia (FYROM)	MK	Country
196	2818	Egypt	Egypt	EG	Country
197	2826	United Kingdom	United Kingdom	GB	Country
198	2834	Tanzania	Tanzania	TZ	Country
199	2840	United States	United States	US	Country
200	2854	Burkina Faso	Burkina Faso	BF	Country
201	2858	Uruguay	Uruguay	UY	Country
202	2860	Uzbekistan	Uzbekistan	UZ	Country
203	2862	Venezuela	Venezuela	VE	Country
204	2876	Wallis and Futuna	Wallis and Futuna	WF	Country
205	2882	Samoa	Samoa	WS	Country
206	2887	Yemen	Yemen	YE	Country
207	2894	Zambia	Zambia	ZM	Country

9 Apêndice B

Comando do Hora de Praticar do capítulo 2

Hora de Praticar 2.1.1

```
1- alunosprof<-c(25280/1751,21328/1186,18432/947,280/29)
escola<-c("Privada","Estadual","Municipal","Federal")
barplot(alunosprof,main="Quatidade de alunos por professor",names.arg=escola,ylab="Quatidade",xlab="Escolas De Ensino Fundamental")
2- barplot(alunosprof,main="Quatidade de alunos por professor",names.arg=escola,xlab="Quatidade",ylab="Escolas De Ensino Fundamental",horiz=TRUE)
3- barplot(alunosprof,main="Quatidade de alunos por professor",names.arg=escola,ylab="Quatidade",xlab="Escolas De Ensino Fundamental",density=60)
4- barplot(alunosprof,main="Quatidade de alunos por professor",names.arg=escola,ylab="Quatidade",xlab="Escolas De Ensino Fundamental",density=50,angle=75)
5- peso=c(42,51,59,64,76)
dias=c("Segundo","Quarto","Sexto","Oitavo","Décimo")
barplot(peso,main="Peso médio dos pintinhos recém nascidos",names.arg=dias,ylab="Peso",xlab="Dias")
6- barplot(peso,main="Peso médio dos pintinhos recém nascidos",names.arg=dias,ylab="Peso",xlab="Dias",col=c("blue", "pink", "yellow", "green", "red"))
7- barplot(peso,main="Peso médio dos pintinhos recém nascidos",names.arg=dias,xlab="Peso",ylab="Dias",col=c("blue", "pink", "yellow", "green", "red"),horiz=TRUE)
8- ocorrencia=c(250,110,85,45,25)
defeito=c("linha ruidosa","linha aberta","alarme","não responde","não toca")
barplot(ocorrencia,main="Defeitos em linhas Telefonicas",names.arg=defeito,xlab="tipo de defeito",ylab="Número de ocorrências",density=80,angle=120)
9- fumaenao=matrix(c(78,133,142,206,197,220,151,412,3300,72,23,15,7,8,9,11), nrow=8,ncol=2,dimnames=list(c("0 -13","14 - 49","50 - 99","100 - 149","150 - 199","200 - 249","250 - 299","300 +"),c("Fumantes","Não Fumantes")))
barplot(fumaenao[,2:1],beside=TRUE,legend.text=rownames(fumaenao), main="Distribuição de nível de cotinina em fumantes e não fumantes",ylab="nível de cotinina",xlab="pessoas",sub="fonte:www.google.com.br")
Hora de Praticar 2.2.1
1- ocorrencia=c(250,110,85,45,25)
names(ocorrencia)=c("linha ruidosa","linha aberta","alarme","não responde","não toca")
porc<-round(ocorrencia*100/sum(ocorrencia),2)
rotulos<-paste(",porc, "%)",sep="")
pie(ocorrencia, main="Defeitos em linhas Telefonicas",labels=rotulos, col=rainbow(7))
legend(1,1, names(ocorrencia), col = rainbow(7), pch=rep(20,6))
2- rotulos<-paste(names(ocorrencia),",",porc, "%)",sep="")
pie(ocorrencia, main="Defeitos em linhas Telefonicas",labels=rotulos, cex=0.8, col=rainbow(7))
3- despesas=c(22200,10500,3530,1040,341)
names(despesas)=c("comida e cigarro", "domésticos", "remédios e saúde", "cuidados pessoais", "educação privada")
porc<-round(despesas*100/sum(despesas),2)
rotulos<-paste(",porc, \"%)",sep="")
pie(despesas, main="Despesas pessoais dos Estados Unidos (em bilhões de dólares)",labels=rotulos, col=rainbow(7))
legend(1,1, names(despesas), col = rainbow(7), pch=rep(20,6))
4-rotulos<-paste(names(despesas),",",porc, "%)",sep="")
pie(ocorrencia, main="Despesas pessoais dos Estados Unidos (em bilhões de dólares)",labels=rotulos, cex=0.8, col=rainbow(7))
Hora de Praticar 2.3.1
1-a) tempm=c(67,72,74,62,56,66,65,59,61,69,74,69,66,68,58,64,66,57,68,62,59,73,61,61,57,58,57,67,81,79,76)
hist(tempm,main="Temperaturas diárias do mês de maio",density=30)
b) temps=c(91,92,93,93,87,84,80,78,75,73,81,76,77,71,71,78,67,76,68,82,64,71,81,69,63,70,77,75,76,68)
hist(temps,main="Temperaturas diárias do mês de setembro",col="violet")
c) graus = (tempm - 32) / 1.8
hist(graus,freq=FALSE,main="Frequência relativa das temperaturas diárias do mês de maio",density=25,col="dark blue")
```

Hora de Praticar 2.4.1

```
1- data(InsectSprays)
attach(InsectSprays)
boxplot(count~spray,main="Boxplot para contagem de insetos tratados com inseticidas",col="lightgray",xlab="tipo de inseticida")
```

```
2- boxplot(count~spray,main="Boxplot para contagem de insetos tratados com inseticidas",col="lightgray", xlab="tipo de inseticida",outline=FALSE)
```

```
3- boxplot(count~spray,main="Boxplot para contagem de insetos tratados com inseticidas",col="lightgray", xlab="tipo de inseticida",boxwex=0.3)
```

```
4- attach(rock)
```

```
x<-peri[area<=mean(area)]
```

```
y<-peri[area>mean(area)]
```

```
peri<-c(x,y)
```

```
area<-c(rep("<=área média",length(x)),rep(">área média",length(y)))
```

```
dados<-data.frame(peri,area)
```

```
boxplot(peri~area,main="Boxplot para perímetro de pedras",xlab="area da pedra")
```

```
5- boxplot(peri~area,main="Boxplot para perímetro de pedras",xlab="area da pedra",varwidth=TRUE)
```

Hora de Praticar 2.5.1

```
1- sapos=c(6,5,10,11,26,16,17,37,18,21,22,15,24,25,29,31,32,13,39,40)
```

```
umid=c(62,24,21,30,34,36,41,48,56,74,57,46,58,61,68,76,79,33,85,86)
```

```
temp=c(31,23,28,30,15,16,24,27,18,10,17,13,25,22,34,12,29,35,26,19)
```

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

```
plot(umid,sapos)
```

```
plot(temp,sapos)
```

```
2- aves<-c(22,28,37,34,13,24,39,5,33,32)
```

```
arvores<-c(25,26,40,30,10,20,35,8,35,28)
```

```
plot(arvores,aves)
```

```
arvores2<-c(6,17,18,11,6,15,20,16,12,15)
```

```
aves2<-c(7,15,12,14,4,14,16,60,13,16)
```

```
points(arvores2,aves2,pch=3,col="blue")
```

```
3- plot(arvores,aves,main="Dispersão entre aves e árvores",xlab="árvores",ylab="aves", col="dark blue",pty='n')
```

```
points(arvores2,aves2,pch=20,col="blue")
```

```
4- internet=c(30, 8, 20, 15, 12, 4, 40, 25, 20, 14, 22, 2, 15, 18, 30, 20, 24, 15, 8, 8)
```

```
at.fisicas=c(2, 10, 5, 5, 8, 15, 0, 4, 5, 10, 2,15, 6, 4, 2, 6, 4, 10, 12, 15)
```

```
plot(internet,at.fisicas,main="Gráfico das horas na internet e praticando atividades física",ylab="Atividade Física",xlab="Internet",col="red")
```

Hora de Praticar 2.6.1

```
1- data(cars)
```

```
attach(cars)
```

```
plot(c(4,25),c(2,120),type="n",main="Gráfico da velocidade e o tempo que o carro demora para parar",ylab="Distância",xlab="Velocidade")
```

```
lines(speed,dist,col="green")
```

```
2- visc=c(47.9, 48.8, 48.6,47.9, 48.1, 48.0,48.6, 48.3, 47.9,48.0, 47.2, 48.3,48.4, 48.9, 48.5,48.1, 48.6, 48.1,48.0, 48.0,48.6, 47.5, 48.3)
```

```
plot(c(0,24),c(47,49),type="n",main="Gráfico de viscosidade para um produto químico",xlab="horas do dia",ylab="viscosidade")
```

```
lines(visc,col=5)
```

```
3-alunos=c(19720,20567,21473,21887,20598,22473,23564)
```

```
anos<-1990:1996
```

```
plot(anos,alunos,type="b",main="Matriculas em milhares de alunos, 1990-1996",ylab="Alunos",xlab="Anos",col="purple")
```

Hora de Praticar 2.7.1

```
1- x<-c(1176, 1191, 1214, 1220, 1205, 1192, 1201, 1190, 1183, 1185)
```

```
mi<-mean(x)
```

```

sigma<-sd(x)
curve(dnorm(x,mean=mi,sd= sigma),lwd=2,from=mi-3*sigma,to=mi+3*sigma)
2- curve(dnorm(x,mean=-1,sd=1),lwd=2,from=-9,to=9)
curve(dnorm(x,mean=-0.5,sd=1),col=2,lwd=2,from=-9,to=9,add=TRUE)
curve(dnorm(x,mean=0,sd=1),col=3,lwd=2,from=-9,to=9,add=TRUE)
curve(dnorm(x,mean=0.5,sd=1),col=4,lwd=2,from=-9,to=9,add=TRUE)
curve(dnorm(x,mean=1,sd=1),col=5,lwd=2,from=-9,to=9,add=TRUE)
legend('topright',legend=c(expression(mi== -1),expression(mi== -0.5), expression(mi== 0), expression(mi== 0.5), expression(mi== 1)),text.col=c(1,2,3,4,5), cex=.85)
title("distribuição normal com variância 1 e médias variando")
3- curve(dnorm(x,mean=0,sd=0),lwd=2,from=-3,to=3)
curve(dnorm(x,mean=0,sd=0.5),col=2,lwd=2,add=TRUE)
curve(dnorm(x,mean=0,sd=1),col=3,lwd=2, add=TRUE)
curve(dnorm(x,mean=0,sd=1.5),col=4,lwd=2,add=TRUE)
curve(dnorm(x,mean=0,sd=2),col=5,lwd=2, add=TRUE)
legend('topright',legend=c(expression(sigma== 0),expression(sigma== 0.5), expression(sigma== 1), expression(sigma== 1.5),expression(sigma== 2)),text.col=c(1,2,3,4,5),cex = .85)
title("distribuição normal com média 0 e desvio padrão variando")
4- pesos<-c(0.9475,0.9705,0.9770,0.9775,0.9860,0.9960,0.9965,0.9975,1.0050,1.0075,1.0100,1.0175,1.0180,1.02000,1.0250)
hist(pesos, freq=F, xlim=c(0.9,1.1))
curve(dnorm(x,mean=1,sd=0.05),col=4,lwd=2, add=TRUE)
curve(dnorm(x,mean=mean(pesos),sd=sd(pesos)),col=2,lwd=2, add=TRUE)
legend('topright',legend=c(expression(mi== 1),expression(sigma== 0.05),expression(mi== 0.9968),expression(sigma== 0.0217)),text.col=c(4,4,2,2),cex=.85)

```

Comando do Hora de Praticar do capítulo 4

Hora de Praticar 4.1.2
1)
p2<-ggplot(data = dados,aes(x=Instituição,y=Frequência,fill=Área)) +
geom_bar(stat='identity',width=0.5) +
labs(x='Instituição',y='Frequência',title="Distribuição de Bolsas por Área do Conhecimento") +
theme_bw()
p2

Comando do Hora de Praticar do Capítulo 5

Hora de Praticar 5.3.1

1)dados=data.frame(ano, tri1,tri2)
L=gvisLineChart(dados, xvar="ano", yvar=c("tri1", "tri2"), options=list(gvis.editor="Edit me!", height=350, width=400))
plot(L)
2)dados=data.frame(nicotina, fumante, n_fumante)
L=gvisLineChart(dados, xvar="nicotina", yvar=c("fumante", "n_fumante"), options=list(gvis.editor="Edit me!", height=350, width=400))
plot(L)
3)dados=data.frame(area, InstA, InstB)
L=gvisLineChart(dados, xvar="area", yvar=c("InstA", "InstB"), options=list(gvis.editor="Edit me!", height=350, width=400))
plot(L)

Hora de Praticar 5.5.1

```
1  
Col1=c( "Aroma A", "Aroma B", "Sabor A", "Sabor B"); Col2=c(7,8.5,5,8.3)  
Gauge <- gvisGauge(data.frame(Col1,Col2), options=list(min=0, max=10, greenFrom=0, greenTo=3, yellowFrom=3, yellowTo=7,  
redFrom=7, redTo=10))  
plot(Gauge)  
2  
Col1=c( "Corredor 1", " Corredor 2", " Corredor 3"); Col2=c(18.5, 19.4, 21.3)  
Gauge <- gvisGauge(data.frame(Col1,Col2), options=list(min=0, max=25)) plot(Gauge)  
3  
Col1=c( "Dia 1", " Dia 2", " Dia 3", " Dia 4", " Dia 5" ); Col2=c(22, 31, 25, 19, 15)  
Gauge <- gvisGauge(data.frame(Col1,Col2), options=list(min=0, max=50, greenFrom=0, greenTo=15, yellowFrom=15, yellowTo=30,  
redFrom=30, redTo=50))  
plot(Gauge)
```

10 Apêndice C

Código de geração do gráfico da capa

```
library("GGally")
library("ggplot2")

data(iris)

colnames(iris)=c("Medida1", "Medida2", "Medida3", "Medida4", "Tipo")
levels(iris$Tipo)<-c("Tip01", "Tip02", "Tip03")
capa=ggpairs( iris, 1:4, mapping = aes(color = Tipo),
lower = list(
    continuous = wrap(ggally_smooth_loess, size = 1, color = "darkblue")),
diag = list(continuous=wrap(ggally_barDiag, color="darkblue")),
upper = list(
    continuous = wrap("cor", size = 4.0, alignPercent = 1) )
)
capa+ggplot2::theme(panel.background = ggplot2::element_rect(fill = "lightgray"))
```

Visualização de dados com o software R

Este livro apresenta os principais pacotes para produção de gráficos com o software R. É um excelente texto para o leitor que quer ter uma visão rápida e introdutória dos comandos para elaboração de gráficos destes pacotes.

Luciane Ferreira Alcoforado é licenciada em Matemática pela UFSM, com doutorado em Engenharia Civil pela UFF. É professora adjunta no departamento de Estatística da UFF. Coordenadora do portal Estatística é com R, vem desenvolvendo materiais didáticos sobre a linguagem R desde 2010. Em 2015 iniciou o projeto do Seminário Internacional de Estatística com R que atualmente encontra-se na sua segunda edição.

Ariel Levy é engenheiro eletricista com doutorado em Economia, ambos pela UFF. É professor adjunto no departamento de Administração da UFF e coordenador do curso de Administração. Vem promovendo a implantação da linguagem R nos cursos de Administração tanto no nível de graduação como de mestrado.

ISBN 978-859229320-8

