

DESENVOLVIMENTO DE UMA APLICAÇÃO INTERATIVA PARA VISUALIZAÇÃO E ANÁLISE DE DADOS¹

Samantha Rico Gonçalves²

Jorge Luis Boeira Bavaresco³

RESUMO

Cada vez mais se buscam maneiras de se aperfeiçoar a visualização e a análise de dados. A linguagem de programação R auxilia, por exemplo, na geração de diversos tipos de gráficos e de informações estatísticas, que são úteis para que se possam extrair informações dos dados. Já o pacote *Shiny* foi desenvolvido para complementar a linguagem R, permitindo a criação de aplicações Web utilizando a linguagem. Este artigo apresenta as etapas do desenvolvimento de uma aplicação interativa utilizando a linguagem R juntamente com o pacote *Shiny*, para facilitar a visualização e a análise de um conjunto de dados. Como resultado da pesquisa obteve-se o protótipo de uma aplicação interativa para visualização e análise de dados, que permitiu que o usuário final interagisse com seus dados de forma dinâmica e facilitada.

Palavras-chave: Facilidade. Estatística. Gráficos. Programação.

1 INTRODUÇÃO

A tecnologia tem papel fundamental em diversas áreas do mundo e está sempre em busca de inovações e formas de facilitar a vida das pessoas. Atualmente, existe uma grande disponibilidade de dados sobre os mais diversos temas, inclusive de forma pública, como é o caso de alguns sites governamentais como o Portal Brasileiro de Dados Abertos e o Instituto de Pesquisa Econômica Aplicada, entre outros, que disponibilizam dados sobre diferentes temas, como economia, educação, geográficos, meteorológicos e assim por diante.

Além disso, podem-se citar as redes sociais, que como Mendes (2011) explica, estão muito presentes na vida de todos, principalmente das novas gerações que já

¹ Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Campus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet, na cidade de Passo Fundo, em 2015.

² Aluna do curso de Tecnologia de Sistemas para Internet no Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense de Passo Fundo (IFSUL). E-mail: sah.rico@gmail.com

³ Orientador, professor do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense de Passo Fundo (IFSUL). E-mail: jlbavaresco@gmail.com.

começam a vida teclando e vivenciando um mundo rápido, instantâneo, com troca de dados a cada instante, convivendo com um enorme volume de informações.

Levando em consideração as afirmações anteriores, percebe-se a importância de se possuir uma forma que facilite a extração de informações dos dados. Assim como Idreos et al. (2015) diz, a exploração de dados tem como objetivo a extração eficaz das informações dos dados, mesmo que o usuário não saiba o que está procurando.

Nesse contexto, o desenvolvimento de uma aplicação que permita a visualização e a análise interativa de um conjunto de dados pode auxiliar os usuários que necessitam extrair, de forma mais facilitada, o conteúdo informacional dos dados.

Para desenvolver a aplicação, foi utilizada a linguagem de programação R para a visualização dos dados, geração de gráficos e informações estatísticas. E o pacote *Shiny* que permitirá o desenvolvimento e a disponibilização da aplicação na Web.

O objetivo principal do trabalho é, desta forma, o desenvolvimento de uma aplicação interativa que torne mais fácil a visualização e a análise de um conjunto de dados.

Portanto, este artigo comporta detalhes sobre o desenvolvimento da aplicação. A seção 2 apresenta o referencial teórico sobre as tecnologias, a 3 descreve como foi desenvolvida a aplicação, a 4 apresenta os resultados obtidos com o desenvolvimento da aplicação e, por fim, a seção 5 traz a conclusão sobre os resultados da pesquisa.

2 REFERENCIAL TEÓRICO

Nesta seção serão explicadas as principais tecnologias utilizadas no desenvolvimento da aplicação de visualização e de análise interativa de dados.

2.1 LINGUAGEM DE PROGRAMAÇÃO R

Matloff et al. (2011) explica que a linguagem de programação R “é uma linguagem de script para manipulação de dados e análise estatística”, ou seja, uma linguagem voltada para fins científicos, matemáticos e estatísticos.

O R é um software livre, desta forma, possui código fonte aberto e pode ser modificado ou implementado com novos recursos por seus usuários contando com grande número de colaboradores das mais diversas áreas. (SOUZA et al., 2008)

Mello et al. (2013) salienta que mesmo que o R não possua suporte técnico, a exemplo do que é oferecido pelas empresas que mantêm um software pago, ainda conta com a colaboração de milhares de usuários, que inclui pessoas renomadas em diversas áreas, como professores e pesquisadores. E é possível conseguir suporte por meio de uma lista de e-mails de perguntas e respostas que tiram dúvidas dos usuários ou por meio do envio de e-mail para um especialista.

A linguagem R, além de todas as características já citadas, possui a capacidade de ler dados compactados (zip, tar.gz etc.) e de ler arquivos de diversos formatos (csv, xls, tab etc). Com o R, é simples obter dados de aplicações em formatos de Web (dados JSON, XML, HTML, etc), além de possuir a capacidade de interagir com diferentes bancos de dados (Postgres, MongoDB etc.). (RADU et al., 2014)

De acordo com Souza et al. (2008), o R é, portanto:

Uma importante ferramenta na análise e na manipulação de dados, com testes paramétricos e não paramétricos, modelagem linear e não linear, análise de séries temporais, análise de sobrevivência, simulação e estatística espacial, entre outros, além de apresentar facilidade na elaboração de diversos tipos de gráficos, no qual o usuário tem pleno controle sobre o gráfico criado.

2.1.1 Histórico

A linguagem R foi criada em 1996 pelo neozelandês Ross Ihaka e pelo canadense Robert Gentleman, inicialmente como um projeto de pesquisa. Foi baseada em duas outras linguagens, sendo elas a linguagem S e a Scheme. (MELLO et al., 2013)

Os seus criadores resolveram disponibilizar o código fonte da linguagem sob os termos da licença GNU (*General Public License*). A partir disso, seus criadores começaram a liberar versões do R, de tempo em tempo, para corrigir erros que seus usuários reportavam, criando-se, então, a lista de e-mail para facilitar o contato com os usuários do R. (IHAKA, 1998)

Segundo Mello et al. (2013), nos dias atuais, o R está sob constante desenvolvimento por um grupo chamado R Core Team e continua em grande crescimento com a colaboração dos seus usuários.

2.1.2 Gráficos

Um dos principais diferenciais da linguagem R é a sua facilidade para a geração de gráficos a partir de um conjunto de dados. Como Mello et al. (2013) diz, a linguagem R é muito poderosa no que diz respeito à confecção de gráficos e afins.

Na tabela 1, é possível visualizar uma relação dos principais gráficos da linguagem R e suas funções.

Tabela 1 – Principais Gráficos da Linguagem R e descrição das suas funções.

Gráficos	Funções
Gráfico Simples ou Gráfico de <i>Plot</i>	O gráfico simples é criado por meio do comando <i>plot()</i> , que é de acordo com Mello et al. (2013) o mais utilizado para a criação de gráficos com o R. Esse comando possui diferentes argumentos sendo possível personalizar o gráfico da forma que o usuário achar necessário.
Gráfico <i>Boxplot</i>	Segundo Mello et al. “Os <i>boxplots</i> são gráficos extremamente úteis, especialmente quando temos uma variável categórica associada aos dados. [...] São várias as informações mostradas no <i>boxplot</i> , cada caixa (box) corresponde aos dados observados.”
Histogramas	Conforme Levine et al. (2008), “um histograma é um gráfico de barras para dados numéricos em grupo no qual as frequências ou porcentagem de cada grupo de dados numéricos são representadas por barras verticais individuais.”

Fonte: Do Autor.

2.1.3 RStudio

O R possui sua própria interface por linha de comando quando instalado. Porém, devido a sua simplicidade, acaba se tornando limitado. Por esse motivo, foi desenvolvido o RStudio, que busca suprir as carências da interface padrão do R.

O RStudio é uma IDE (*Integrated Development Environment*) gratuita que possui versões para diferentes sistemas operacionais. Essa IDE oferece uma interface que integra o R e um avançado editor de textos voltado para edição de comandos. (MELLO et al., 2013)

2.2 PACOTE SHINY

Segundo Radu et al. (2014), o R não serve apenas para análise quantitativa, mas é usado também para construir aplicativos de desktop e aplicações Web. E foi para permitir o desenvolvimento de aplicações Web interativas, com a utilização do R, que surgiu o pacote *Shiny*.

O pacote *Shiny* permite que os usuários transformem suas análises em aplicações Web interativas e que podem ser acessadas por diversos outros usuários. E o mais interessante é que, para desenvolver a aplicação, não é necessário possuir um conhecimento prévio em HTML, CSS ou Javascript, já que a programação é realizada toda em R. (RADU et al., 2014)

As aplicações criadas com o *Shiny* são divididas em dois arquivos, o *ui.R* e o *server.R*, que são scripts que fazem parte do pacote *Shiny*. O script *ui.R* controla o layout e a aparência da aplicação, enquanto o script *server.R* contém as instruções que o computador precisa para construir a aplicação. (RSTUDIO INC., 2014)

O pacote *Shiny* se tornou uma alternativa, portanto, para os usuários que necessitam desenvolver aplicações Web interativas de forma facilitada, utilizando a linguagem R, podendo criar as mais diversas aplicações, para as mais diversas funções.

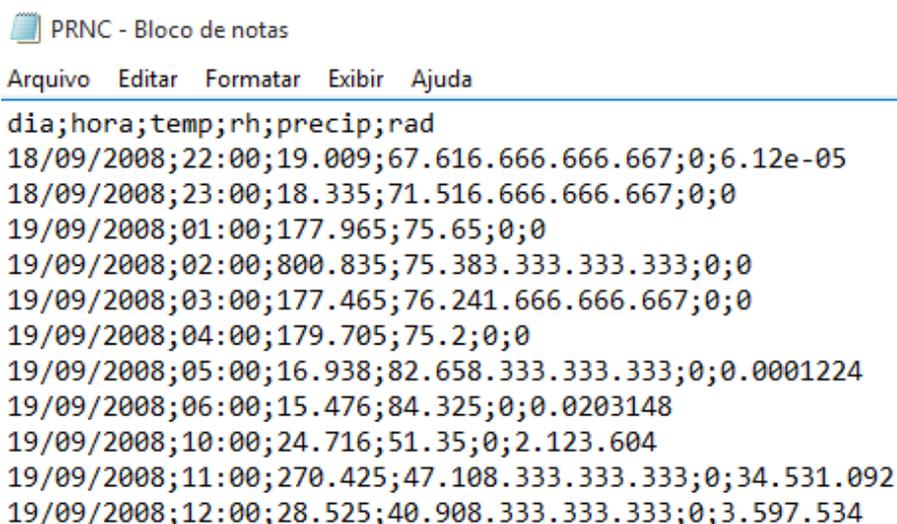
2.3 FORMATO DE DADOS CSV

Segundo Recipi (2004), o formato CSV (*Comma Separated Values*) é usado frequentemente para trocar dados entre aplicações diferentes.

Algumas características dos arquivos no formato CSV são citadas por Shafranovich (2005), como sua configuração, onde explica que cada registro se localiza em linhas separadas, delimitadas por uma quebra de linha. Pode haver um cabeçalho opcional na primeira linha do arquivo. Cada campo pode ou não estar com aspas duplas.

Na figura 1, é possível visualizar um exemplo de arquivo no formato CSV e como seus campos são comumente dispostos.

Figura 1 – Exemplo de arquivo CSV



```
PRNC - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
dia;hora;temp;rh;precip;rad
18/09/2008;22:00;19.009;67.616.666.666.667;0;6.12e-05
18/09/2008;23:00;18.335;71.516.666.666.667;0;0
19/09/2008;01:00;177.965;75.65;0;0
19/09/2008;02:00;800.835;75.383.333.333.333;0;0
19/09/2008;03:00;177.465;76.241.666.666.667;0;0
19/09/2008;04:00;179.705;75.2;0;0
19/09/2008;05:00;16.938;82.658.333.333.333;0;0.0001224
19/09/2008;06:00;15.476;84.325;0;0.0203148
19/09/2008;10:00;24.716;51.35;0;2.123.604
19/09/2008;11:00;270.425;47.108.333.333.333;0;34.531.092
19/09/2008;12:00;28.525;40.908.333.333.333;0;3.597.534
```

Fonte: Do autor.

3 DESENVOLVIMENTO

Nesta seção, será apresentada a metodologia utilizada para o desenvolvimento da aplicação, definindo os requisitos necessários para que a aplicação cumpra com seus objetivos. Além disso, serão demonstrados detalhes sobre o desenvolvimento da aplicação, apresentando sua estrutura e funcionamento.

3.1 METODOLOGIA

Os profissionais têm dificuldades em encontrar aplicações, principalmente que utilizem tecnologias de software livre e que permitam a interação com os dados de uma forma fácil e interativa. Por essa razão, a aplicação deve atender e respeitar a alguns requisitos e funcionalidades específicas, para cumprir o objetivo de facilitar a interação dos profissionais com os conjuntos de dados dela.

Os requisitos foram levantados com base nas principais necessidades dos usuários. O primeiro requisito é que a aplicação permita o envio de arquivos de dados no formato *csv* ou *txt*, deve possibilitar que o usuário informe as características do arquivo, como por exemplo, qual é o separador dos dados (ponto, vírgula, ponto e vírgula), se os dados possuem cabeçalho ou não, entre outras informações pertinentes.

Além disso, é de suma importância para o cumprimento do seu objetivo, exibir os dados do arquivo, enviado na tela, para que o usuário possa visualizá-los de forma interativa.

O usuário, também, deverá ter a opção de gerar diferentes tipos de gráficos a partir dos dados do arquivo enviado, tais como *boxplot*, histograma e *plot*, além de realizar alguns cálculos básicos de estatística (média aritmética, mediana, quartis), para que se possa obter uma melhor análise sobre as informações.

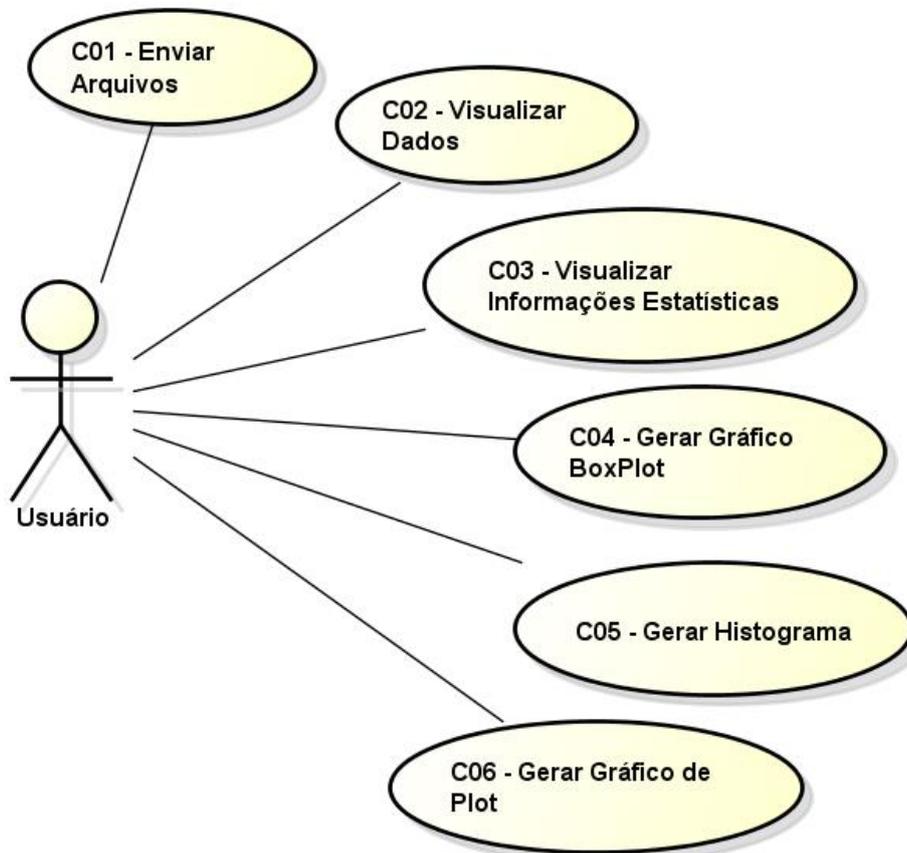
A partir dos requisitos, portanto, será desenvolvida a aplicação utilizando a linguagem de programação R e sua IDE (*Integrated Development Environment*) o RStudio, juntamente com o pacote *Shiny*, para que seja possível disponibilizar a aplicação na Web.

3.1.1 Diagrama de Caso de Uso

“O diagrama de caso de uso é o mais geral e informal da UML (*Unified Modeling Language*)”, de acordo com Guedes (2011). Esse tipo de diagrama é utilizado para modelagem de aspectos mais dinâmicos dos sistemas, utilizado para descrever seu comportamento, sendo composto pelo relacionamento dos casos de uso e seus atores. (BOOCH, 2012)

Na figura 2, é possível visualizar o diagrama de caso de uso da aplicação. Onde são apresentadas as principais interações que o usuário poderá realizar.

Figura 2 – Diagrama de Caso de Uso



Fonte: Do autor.

3.2 SOBRE A APLICAÇÃO

Como já explicado, anteriormente, a aplicação foi desenvolvida utilizando a linguagem de programação R, junto com seu pacote *Shiny*. Antes de mais nada, é necessário entender como a aplicação está estruturada. Na figura 3, é possível visualizar a estrutura da aplicação, que está dividida em dois arquivos principais o “*ui.R*” e o “*server.R*”.

Figura 3 – Arquivos da Aplicação



Fonte: Do autor.

O arquivo “*ui.R*” contém o código referente à interface da aplicação e o arquivo “*server.R*” contém o código que realiza a comunicação para o funcionamento da aplicação.

Na figura 3, ainda, é possível ver uma pasta “*www*”. Essa é uma pasta especial, já que o *Shiny* faz com que todo arquivo contido nela fique disponível para o navegador do usuário. Dessa forma, torna-se o local ideal para guardar arquivos CSS, imagens e outros arquivos necessários para que o *Shiny* construa a aplicação.

Para cumprir com os requisitos da aplicação, a primeira função desenvolvida foi o envio de um arquivo de dados pelo usuário. Na linguagem R, o comando que realiza a leitura de dados é o *read.table()*, na figura 4, é possível visualizar como foi realizada a leitura de dados na aplicação.

Figura 4 – Código Leitura de Dados

```
##lendo os dados
dados <- reactive ({
  inFile <- input$file1

  if (is.null(inFile))
    return(NULL)

  withProgress(message = 'Carregando os dados', value = 0,{
    incProgress(1, detail = paste("Carregando os dados"))
    incProgress(2, detail = paste("Calculando"))
    read.table(inFile$datapath, header=input$header,
              quote=input$quote, sep=input$sep, fill= T)
  })
})
```

Fonte: Do autor.

A função `reactive()` foi utilizada porque possibilita que ela seja utilizada diversas vezes durante a aplicação, ou seja, essa função poderá ser reativada sempre que necessário.

Para que a leitura de dados se torne interativa, o usuário terá a opção de informar alguns parâmetros, como *header* (cabeçalho), *quote* (aspas), *sep* (separador). A figura 5 mostra o código da interface onde é gerada uma *sidebar*, que é a barra fixada na lateral esquerda da aplicação, onde estão as opções de características do arquivo de dados.

Figura 5 – Interface de opções dos dados

```
sidebarPanel(
  conditionalPanel(
    condition="input.conditionedPanels==1",
    fileInput('file1', 'Escolha o arquivo',
      accept=c('text/csv',
        'text/comma-separated-values,text/plain',
        '.csv', '.txt')),
    tags$hr(),
    checkboxInput('header', 'Cabeçalho', TRUE),
    radioButtons('sep', 'Separador',
      c('Sem separador'='',
        vírgula=',',
        'Ponto e Vírgula'=';',
        Tabulação='\t'),
      ','),
    radioButtons('quote', 'Aspas',
      c(Nenhuma='',
        'Aspas Duplas'='\"',
        'Aspas Simples'='\"'),
      '\"')
  ),
),
```

Fonte: Do autor.

A figura 6 exibe a tela principal da aplicação e nela é possível visualizar as possibilidades de escolha de características que o usuário poderá informar de acordo com o arquivo enviado. Após o envio do arquivo, os dados são exibidos na tela, onde os resultados são paginados, ou seja, os dados são dispostos em páginas, cada página contendo uma determinada quantidade de dados para facilitar

a navegação do usuário. Ainda, é possível efetuar consultas utilizando filtros sobre as colunas e buscar dados específicos por meio da busca, alcançando, assim, a interatividade da visualização de dados.

Figura 6 – Características dos Dados

The screenshot displays a web interface for data visualization. On the left, a sidebar titled 'Escolha o arquivo' (Choose the file) shows a file upload section with 'Choose File' and 'Upload complete' buttons, and a configuration section for 'Separador' (Separator) and 'Aspas' (Quotes). The main area has tabs for 'Visualização', 'Gráfico BoxPlot', 'Histograma', 'Gráfico de Plot', and 'Sumário'. Below the tabs, there's a 'Show 10 entries' dropdown and a 'Search:' input field. A table displays data with columns 'Data' and 'Dívida.externa...U!'. At the bottom, there are input fields for column selection and a pagination control showing 'Showing 1 to 10 of 59 entries' and a page navigation bar with 'Previous', '1', '2', '3', '4', and '5' buttons.

Data	Dívida.externa...U!
1956	2736
1957	2491
1958	2870
1959	3160
1960	3738
1961	3291
1962	3533
1963	3612
1964	3294
1965	3823

Fonte: Do autor.

É necessário destacar que a aplicação foi desenvolvida, dessa maneira, para que o usuário possa enviar e visualizar seus arquivos de dados de forma interativa, devido ao fato de que os dados podem possuir diferentes tipos de configurações.

Outra parte muito importante da aplicação é a geração de gráficos. A aplicação permite a geração de três tipos diferentes de gráficos que são o *boxplot*, o histograma e o *plot*, sendo que cada um deles possui uma implementação diferente.

Primeiramente, para que seja possível a geração de gráficos de forma dinâmica, faz-se necessário exibir para o usuário as colunas existentes no conjunto de dados, possibilitando que ele selecione qual coluna utilizar nos eixos x e y dos gráficos, como se visualiza no código da figura 7.

Figura 7 – Gerando opções da *combobox*

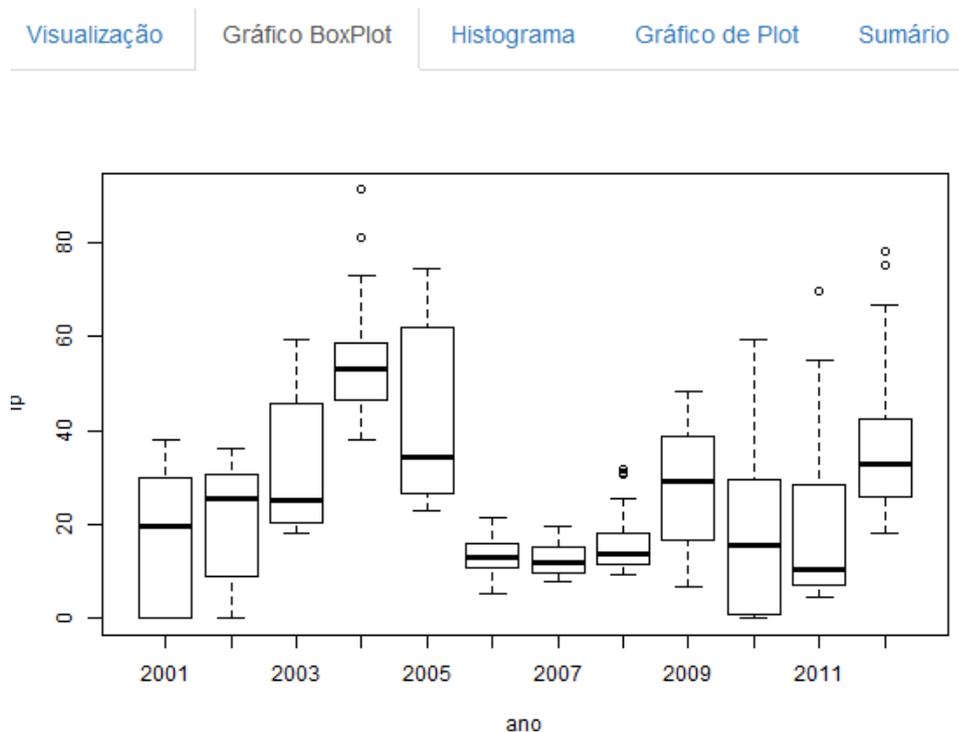
```
##### GERANDO AS OPCOES DOS GRÁFICOS #####

#Gerando opções para o gráfico boxplot
output$boxplotselect1 <- renderUI({
  selectInput(inputId = "boxplot1",
             label = "y:",
             choices = c(dadosChoice()))
})
output$boxplotselect2 = renderUI({
  selectInput(inputId = "boxplot2",
             label = "x:",
             choices = c(dadosChoice()))
})
```

Fonte: Do autor.

A exibição dinâmica das colunas dos dados foi realizada para os outros tipos de gráficos, também. Após gerar as opções para o usuário selecionar, foi gerado o gráfico em si, utilizando essas informações. Na figura 8, é possível visualizar como é exibido o gráfico do tipo *boxplot*.

Figura 8 – Exibindo o gráfico *boxplot*



Fonte: Do autor.

Para que a exibição do gráfico fosse possível, utilizou-se a função *renderPlot()*, que permite renderizar o gráfico na tela. No código fonte, mostrado na figura 9, é chamada a função de leitura de dados, buscam-se os nomes das colunas dos dados e então, por meio da função *boxplot()*, gera-se o gráfico utilizando as informações escolhidas pelo usuário. Dessa forma, a geração de gráficos se torna dinâmica.

Figura 9 – Gerando o gráfico *boxplot*

```
#gerando o grafico boxplot
output$boxplot <- renderPlot({
  dados <- dados()
  validate(
    need(dados != "", "\n Por favor selecione um arquivo.")
  )
  nomes <- names(dados())

  boxplot(dados[,as.integer(input$boxplot1)]~
          dados[,as.integer(input$boxplot2)],
          ylab=nomes[as.integer(input$boxplot1)],
          xlab=nomes[as.integer(input$boxplot2)])
})
```

Fonte: Do autor.

Para a geração dos outros tipos de gráficos, o histograma e o *plot*, utilizou-se o mesmo segmento. A exibição de ambos foi realizada da mesma forma que a do gráfico *boxplot*. O usuário seleciona as opções das colunas e o gráfico é exibido. A figura 10, mostra como é a exibição dos gráficos.

Figura 10 – Exibição dos gráficos *plot* e histograma



Fonte: Do autor.

Para gerar os gráficos histograma e *plot* foram utilizadas, respectivamente, as funções *hist()* e *plot()*, tendo como parâmetros as informações escolhidas pelo usuário. O código fonte de como foram gerados os gráficos pode ser visto na figura 11.

Figura 11 – Gerando os gráficos histograma e *plot*

```
#gerando o grafico plot
output$plot <- renderPlot({
  dados <- dados()
  validate(
    need(dados != "", "\n Por favor selecione um arquivo.")
  )
  nomes <-names(dados())
  plot(dados[,as.integer(input$plot1)]~
        dados[,as.integer(input$plot2)],
        ylab=nomes[as.integer(input$plot1)],
        xlab=nomes[as.integer(input$plot2)])
})
#gerando o histograma
output$histograma <- renderPlot({
  dados <- dados()
  validate(
    need(dados != "", "\n Por favor selecione um arquivo.")
  )
  nomes <-names(dados())
  hist(main=input$tituloH,dados[,as.integer(input$histograma)],
        xlab = nomes[as.integer(input$histograma)],
        ylab = "frequências absolutas")
})
```

Fonte: Do autor.

Outra parte muito importante da aplicação é a geração de informações estatísticas como média, mediana, entre outras. Para isso, utiliza-se a função `summary()` que gera, automaticamente, essas informações. Na figura 12, é possível ver como o sumário foi criado.

Figura 12 – Gerando o sumário

```
#gerando as informações do sumário

output$summary <- renderPrint({
  dados <- dados()
  summary(dados)
})

)|
```

Fonte: Do autor.

É relevante salientar a importância do sumário para que a aplicação cumpra com o seu objetivo, já que é por meio dessas informações, além dos gráficos, que o usuário poderá realizar uma melhor análise sobre as informações e encontrar possíveis inconsistências nelas. A figura 13 mostra como as informações são exibidas na aplicação. No caso, para cada coluna do conjunto de dados, está sendo exibida as suas informações de sumário (média, mediana, menor valor, maior valor, primeiro quartil e terceiro quartil).

Figura 13 – Exibição do sumário

Visualização	Gráfico BoxPlot	Histograma	Gráfico de Plot	Sumário
Cultivar	Estadio	Desfolha	Repeticao	Rendimento
Ativa:48	Min. :1.0	Min. : 0.00	Min. :1.00	Min. : 341.6
	1st Qu.:1.0	1st Qu.: 8.00	1st Qu.:1.75	1st Qu.: 925.0
	Median :1.5	Median :25.00	Median :2.50	Median :2573.9
	Mean :1.5	Mean :29.17	Mean :2.50	Mean :2221.0
	3rd Qu.:2.0	3rd Qu.:50.00	3rd Qu.:3.25	3rd Qu.:3390.5
	Max. :2.0	Max. :67.00	Max. :4.00	Max. :3586.1

Fonte: Do autor.

Em relação à interface, cada função como visualização, geração de gráfico e sumário, é dividida em abas separadas, em que o usuário pode navegar de acordo com sua necessidade, como mostra a figura 14.

Figura 14 – Divisão em abas

Visualização	Gráfico BoxPlot	Histograma	Gráfico de Plot	Sumário
Show	10	▼	entries	
Cultivar	Estadio	Desfolha	Repeticao	
Ativa	1	0	1	
Ativa	1	0	2	
Ativa	1	0	3	
Ativa	1	0	4	
Ativa	1	8	1	
Ativa	1	8	2	
Ativa	1	8	3	

Fonte: Do autor.

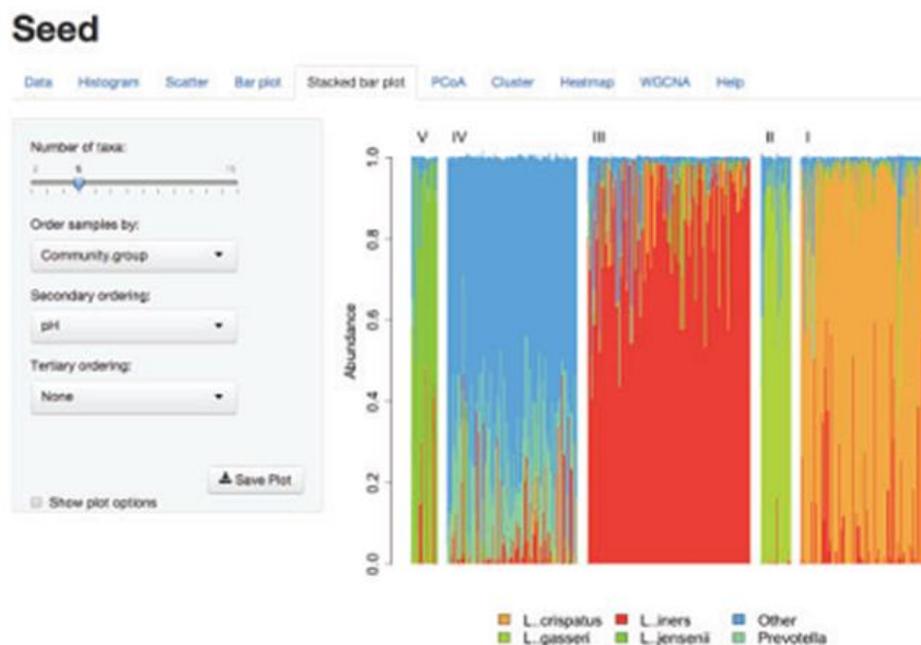
A aplicação foi testada, localmente, por intermédio do RStudio, mas o *Shiny* permite que a aplicação seja disponibilizada na Web, por meio de seu próprio servidor: o *Shiny* Server. O servidor *Shiny* possibilita que o usuário possua mais de uma aplicação hospedada nele, cada uma com uma URL própria.

Essa é uma visão geral sobre o desenvolvimento da aplicação. O importante é que se buscou tornar a aplicação interativa o bastante para que o usuário possa visualizar e analisar seus dados de uma forma facilitada.

3.3 APLICAÇÕES SIMILARES

Existem alguns trabalhos similares ao que está sendo desenvolvido. Entre eles, é possível citar o “*Seed: a user-friendly tool for exploring and visualizing microbial community data*” (Seed: Uma ferramenta amigável para exploração e visualização de dados da comunidade microbiana). Neste trabalho, é desenvolvida uma aplicação que permite que pesquisadores da comunidade microbiana explorem visualmente os dados. (BECK, 2014). Na figura 15, é possível visualizar a interface principal da aplicação Seed.

Figura 15 - Interface da Aplicação Seed



Fonte: BECK (2014)

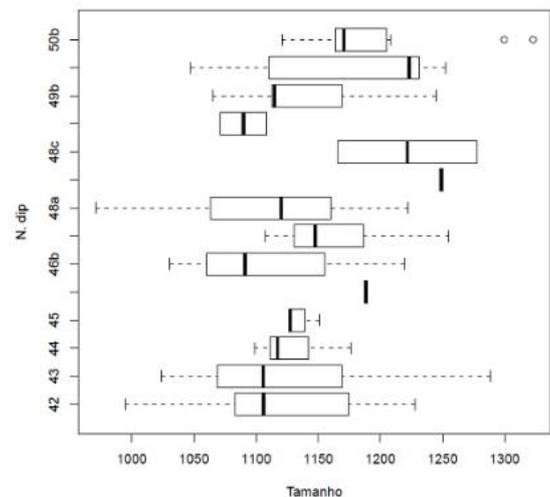
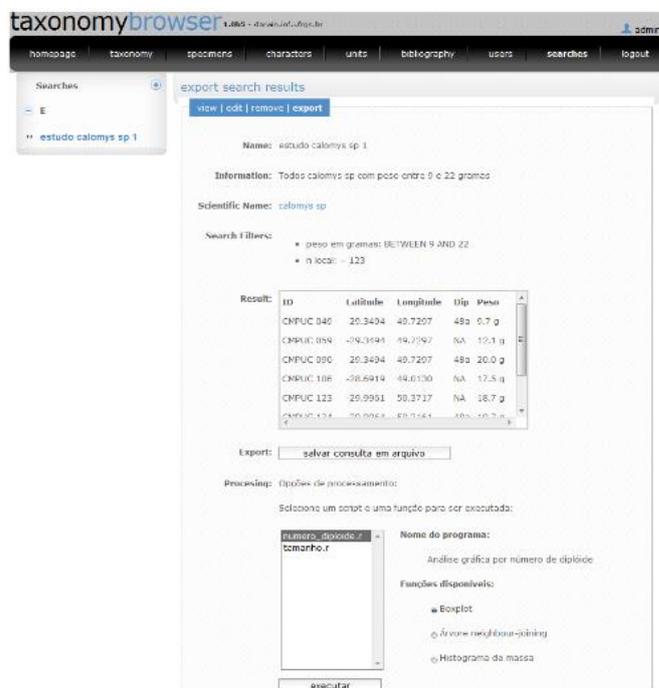
Outro trabalho relacionado, que pode ser citado, é o artigo “Integrando visualização e análise de dados em sistema de gerenciamento de dados de biodiversidade” realizado na Universidade Federal do Rio Grande do Sul.

Nesse trabalho, foi desenvolvido um sistema chamado de TaxonomyBrowser que tem como objetivo, conforme Cañete (2010), “auxiliar os biólogos no gerenciamento de dados a respeito de espécimes por eles coletados, assim como exemplares já catalogados em coleções de museus”.

Nessa aplicação, os criadores utilizaram o Sistema R, para, segundo Cañete (2010), permitir “a execução de consultas avançadas para análise de dados por parte dos biólogos. As consultas otimizadas são processadas por meio de aplicações externas, que são integradas no sistema TaxonomyBrowser de forma transparente ao usuário”

Na figura 16, é possível visualizar a interface de análise do sistema TaxonomyBrowser, na qual foram utilizados scripts R para o desenvolvimento.

Figura 16 - Interface de Análise do Sistema TaxonomyBrowser



Fonte: CAÑENE (2010).

A aplicação “BoxPlotR: a web-tool for generation of box plots” pode ser citada. Foi desenvolvida, utilizando a linguagem R e o pacote Shiny, em parceria com a

Natural Method, e tem como objetivo a visualização de dados e a geração de gráficos customizados do tipo boxplot. Na figura 17 é possível visualizar a interface principal da aplicação.

Figura 17 - Interface da Aplicação BoxplotR

BoxPlotR: a web-tool for generation of box plots

Introduction

About Data upload Data visualization Figure legend template News FAQ

This application was developed with Nature Methods as described in this [editorial](#) and this [blog entry](#). Nature methods also dedicated a [Points of View](#) and a [Points of Significance](#) column to box plots. We hope that you find the [BoxPlotR](#) useful and we welcome suggestions for additional features by our users. We would like to thank everyone who has made constructive suggestions so far. We will document the addition of new features in the News tab.

This application allows users to generate customized [box plots](#) in a number of variants based on their data. A data matrix can be uploaded as a file or pasted into the application. Basic box plots are generated based on the data and can be modified to include additional information. Additional features become available when checking that option. Information about sample sizes can be represented by the width of each box where the widths are proportional to the square roots of the number of observations n . Notches can be added to the boxes. These are defined as $\pm 1.58 \cdot IQR / \sqrt{n}$ which gives roughly 95% confidence that two medians are different. It is also possible to define the whiskers based on the ideas of Spear and Tukey. Additional options of data visualization (violin and bean plots) reveal more information about the underlying data distribution. Plots can be labeled, customized (colors, dimensions, orientation) and exported as eps, pdf and svg files.

BoxPlotR code can be run locally via [GitHub](#). You can also download and install it as a virtual machine (see [GitHub](#) and FAQs for details).

Software references

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna (2013)

RStudio and Inc. *shiny: Web Application Framework for R*. R package version 0.5.0 (2013)

Adler, D. *voplot: Violin plot*. R package version 0.2 (2005)

Eklund, A. *beeswarm: The bee swarm plot, an alternative to stripchart*. R package version 0.1.5 (2012)

Kampstra, P. *Beanplot: A Boxplot Alternative for Visual Comparison of Distributions*. Journal of Statistical Software, Code Snippets 28(1): 1-9 (2008)

Neuwirth, E. *RColorBrewer: ColorBrewer palettes*. R package version 1.0-5. (2011)

Further references

Hadley Wickham and Lisa Stryjewski: [40 years of boxplots](#)

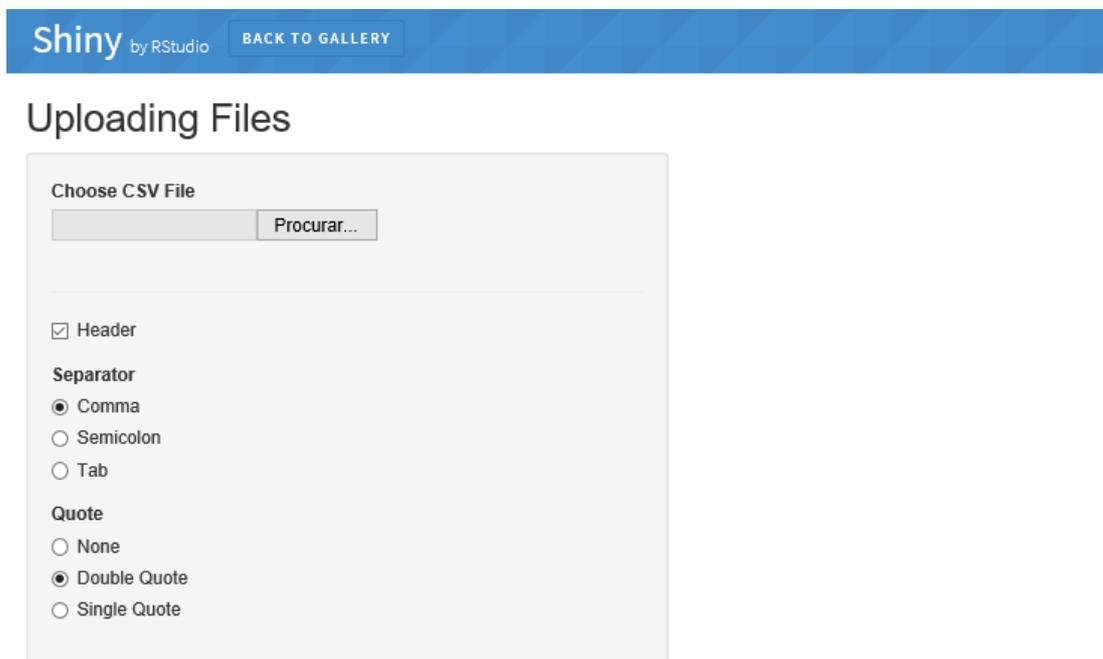
Kristin Potter: [Methods for Presenting Statistical Information: The Box Plot](#)

This application was created by the [Tyers](#) and [Rappsilber](#) labs. Please send bugs and feature requests to Michaela Spitzer ([michaela.spitzer\(at\)gmail.com](mailto:michaela.spitzer(at)gmail.com)) and Jan Wildenhain ([jan.wildenhain\(at\)gmail.com](mailto:jan.wildenhain(at)gmail.com)). This application uses the [shiny package from RStudio](#).

Fonte: Do Autor.

Por fim, outra aplicação que pode ser citada, é a “Shiny – File Upload”⁴ que faz parte da galeria de exemplos de aplicações que o site oficial do pacote Shiny disponibiliza. A aplicação foi desenvolvida para realizar o upload de arquivos de dados e mostrar os dados do arquivo na tela para o usuário. A figura 18, mostra a interface principal da aplicação.

⁴ <http://shiny.rstudio.com/gallery/file-upload.html>

Figura 18 - Interface da Aplicação Shiny - File Upload

Shiny by RStudio [BACK TO GALLERY](#)

Uploading Files

Choose CSV File

Header

Separator

Comma

Semicolon

Tab

Quote

None

Double Quote

Single Quote

Fonte: Do Autor.

4 RESULTADOS OBTIDOS

Ao fim do desenvolvimento da aplicação, atingiu-se o seu principal objetivo que é a interatividade, já que o usuário poderá analisar e visualizar os seus dados de uma forma mais facilitada do que se tivesse que apenas lê-los de uma maneira convencional ou se tivesse que recorrer à utilização da programação.

A aplicação se torna interativa a partir do momento que permite que o usuário interaja de forma direta e dinâmica com seus dados. Desde o momento da visualização dos dados, que permite a realização de consultas, divisão dos dados em colunas e divisão dos dados em páginas para facilitar a navegação.

Além disso, os gráficos são gerados de forma dinâmica, com a possibilidade da seleção das colunas do conjunto de dados para os valores dos eixos x e y dos gráficos. E um sumário com informações como média, menor e maior valor dos dados, entre outras. Todos esses recursos são encontrados na aplicação.

Uma das desvantagens da aplicação é que existe certa dificuldade na leitura de dados com tamanhos muito grandes ou com uma má formatação, o que acaba

acarretando alguns erros, como a lentidão na hora de visualizar os dados, como gerar os gráficos e o sumário.

Se comparada com algumas aplicações similares, também desenvolvidas utilizando o pacote *Shiny*, pode-se afirmar que a aplicação de visualização e análise de dados traz alguns recursos adicionais, tornando-se mais completa para a função que se propõe. Na tabela 2, é realizado um comparativo entre as aplicações.

Tabela 2 – Comparativo com Aplicações Similares.

Aplicação	Upload de Arquivo	Visualização dos Dados	Geração de diferentes gráficos	Download dos gráficos	Geração de Informações Estatísticas
<i>Shiny – File Upload</i>	Sim	Sim, mas não possui filtros e consultas	Não	Não	Não
<i>BoxPlotR: a web-tool for generation of box plots</i>	Sim	Sim, mas não possui filtros e consultas	Não, apenas do tipo <i>boxplot</i>	Sim	Não
Aplicação de Visualização e Análise de Dados	Sim	Sim, visualização dinâmica com filtros e consultas	Sim	Sim	Sim

Fonte: Do Autor.

Para validar que a aplicação cumpre com o que propõe, foram realizados alguns testes com conjuntos de dados. Utilizaram-se dados⁵ de saída de um modelo de simulação de doenças, no caso, a brusone no trigo. Na figura 19, pode-se visualizar como o conjunto de dados utilizado está estruturado.

⁵ Dados fornecidos pelo professor Jorge Luís Boeira Bavaresco.

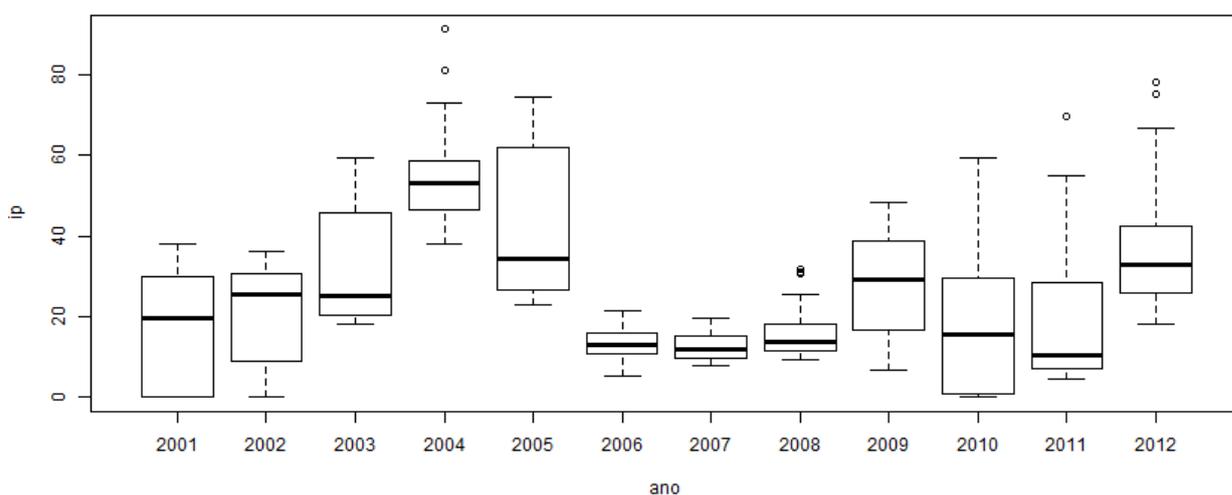
Figura 19 – Dados Brusone no Trigo

treatment	semeadura	ano	dia	city	latitude	longitude	cultivar	heading_date	ip	incidence
184933	2001-03-26	2001	26-3	CIANORTE	-23.4	-52.35	BRS-LOURO	2001-07-27	0	0
184934	2001-03-31	2001	31-3	CIANORTE	-23.4	-52.35	BRS-LOURO	2001-08-01	0	0
184935	2001-04-05	2001	5-4	CIANORTE	-23.4	-52.35	BRS-LOURO	2001-08-06	0	0
184936	2001-04-10	2001	10-4	CIANORTE	-23.4	-52.35	BRS-LOURO	2001-08-11	0	0
184937	2001-04-15	2001	15-4	CIANORTE	-23.4	-52.35	BRS-LOURO	2001-08-16	0	0

Fonte: Do autor.

Nessa primeira análise, foi selecionada a coluna ip (inóculo potencial) e o ano, utilizando o gráfico *boxplot*, para visualizar a quantidade de inóculo que ocorreu a cada ano, como mostra a figura 20.

Figura 20 – Analisando dados por meio de gráficos

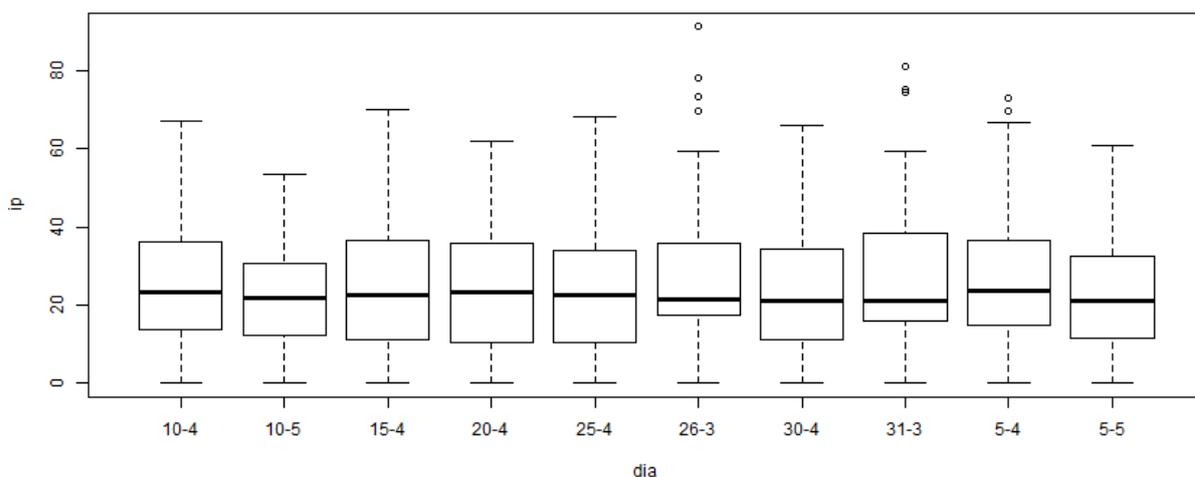


Fonte: Do autor.

Por meio desta análise, é possível que o usuário possa ter uma melhor noção sobre os anos em que a incidência da doença foi maior ou menor, auxiliando aos pesquisadores a extração da informação sobre o comportamento da doença ao longo dos anos.

Outro bom exemplo de análise, ainda utilizando os dados da brusone no trigo, é selecionar a coluna do inóculo potencial e dos dias de semeadura. Dessa forma, o usuário poderá analisar qual época de semeadura é mais suscetível à doença, como mostra a figura 21.

Figura 21 – Analisando dados por meio de gráficos



Fonte: Do autor.

Os resultados obtidos, portanto, estão de acordo com o que se propôs alcançar com o desenvolvimento da aplicação.

5 CONSIDERAÇÕES FINAIS

O objetivo desse projeto era o desenvolvimento, utilizando a linguagem de programação R e o pacote *Shiny*, de uma aplicação interativa capaz de aperfeiçoar e facilitar a visualização e a análise de um conjunto de dados. O objetivo do projeto foi atingido por meio da construção de uma aplicação que permite que o usuário interaja com seus dados de forma dinâmica e facilitada.

É interessante salientar a notável facilidade com que é possível desenvolver uma aplicação Web utilizando a linguagem de programação R e seu pacote *Shiny*. Os resultados, que foram obtidos por meio do desenvolvimento, não teriam sido possíveis sem os diversos recursos que essas tecnologias disponibilizam. Em

termos de conhecimento, pode-se afirmar que aprender a linguagem R pode ser de muita utilidade, quando se trata de visualização e de análise de dados.

A maior dificuldade, encontrada durante o desenvolvimento, foi a busca por formas de tornar a aplicação interativa, já que, em termos de desenvolvimento, seria mais fácil, por exemplo, desenvolver uma aplicação para um conjunto de dados específico do que para dados de diversos tipos. Pois não se sabe a forma como os dados, enviados para a aplicação, estão estruturados, o que gera certa dificuldade para o desenvolvimento.

Como proposta para trabalhos futuros, seria interessante, por exemplo, a realização da integração da aplicação com o banco de dados, já que a linguagem R permite esse tipo de interação, para que, assim, o usuário possa buscar seus dados diretamente do banco de dados. E, também, pode ser realizada a adição de novos recursos e funcionalidades à aplicação existente, como novos gráficos e informações estatísticas.

ABSTRACT

Increasingly seek ways to improve visualization and data analysis. The R programming language supports, for example, the generation of several types of graphs and statistical informations, useful to extract informations from data. Shiny package was developed to complement the R language, allowing you to create Web applications using the language. This paper presents the stages of development of an interactive application using the R language with R Shiny package, to facilitate the visualization and analysis of a data set.

Keywords: Facility. Statistic. Charts. Programming.

REFERÊNCIAS

BECK, Daniel, DENNIS, Christopher; FOSTER, James. (2014) "*Seed: a user-friendly tool for exploring and visualizing microbial community data*". Department of Biological Sciences, University of Idaho, Moscow, v. 31, n. 4, p. 602-603, outubro.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *UML: Guia do Usuário*. 2. ed. Rio de Janeiro: Elsevier Editora Ltda., 2012.

CAÑETE, Samantha et al. *Integrando visualização e análise de dados em sistema de gerenciamento de dados de biodiversidade*. Disponível em <<http://goo.gl/qlUqhe>> Acesso em: 12 jun. 2015.

GUEDES, Gilleanes. *UML 2 - Uma Abordagem Prática*. 2. ed. São Paulo: Novatec Editora Ltda, 2011.

IDREOS S; PAPAEMMANOUIL O; Chaudhuri S. (2015) “*Overview of Data Exploration Techniques*”. ACM SIGMOD International Conference on Management of Data, Tutorial. Melbourne, Austrália.

IHAKA, Ross. *R: Past and Future History*. Disponível em <<http://goo.gl/9N94cc>> Acesso em: 11 jun. 2015.

LEVINE, David; STEPHAN, David; KREHBIEL, Timothy; BERENSON, Mark. *Estatística: Teoria e Aplicações*. 5. ed. Rio de Janeiro: LTC Editora, 2008.
MATLOFF, Norman. (2011) “*The Art of R Programming: A tour of statistical software design*”. 1. ed. San Francisco: No Starch Press.

MELLO, Marcio; PETERNELLI, Luiz. “*Conhecendo o R: Uma visão estatística*”. (2013) 1. ed. Viçosa: Editora UFV.

MENDES, Alexandre. (2011). *As redes sociais e sua influência na sociedade*. Disponível em <<http://goo.gl/6P5YSB>> Acesso em: 29 out. 2015.

RADU, Marius; MURESAN, Ioana; NISTOR, Răzvan. (2014) “*Using R To Get Value Out Of Public Data*”. Disponível em < <http://goo.gl/GQQckw>> Acesso em: 10 jun. 2015.

REPICI, J. *HOW-TO: The Comma Separated Value (CSV) File Format*. Disponível em <<http://goo.gl/PFDgH>> Acesso em: 07 jul. 2015.

RSTUDIO INC. (2014). Shiny: Web Application Framework for R. Disponível em <<http://goo.gl/EDyxyt>> Acesso em: 11 jun. 2015.

SHAFRANOVICH, Yakov. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Disponível em <<http://goo.gl/GwxKnE>> Acesso em: 07 jul. 2015.

SOUZA, Emanuel Fernando Maia; PETERNELLI, Luiz Alexandre; DE MELLO, Márcio Pupin. (2008) “*Software Livre R: aplicação estatística*”. Disponível em <<http://goo.gl/xQqK27>> Acesso em: 08 jun. 2015.