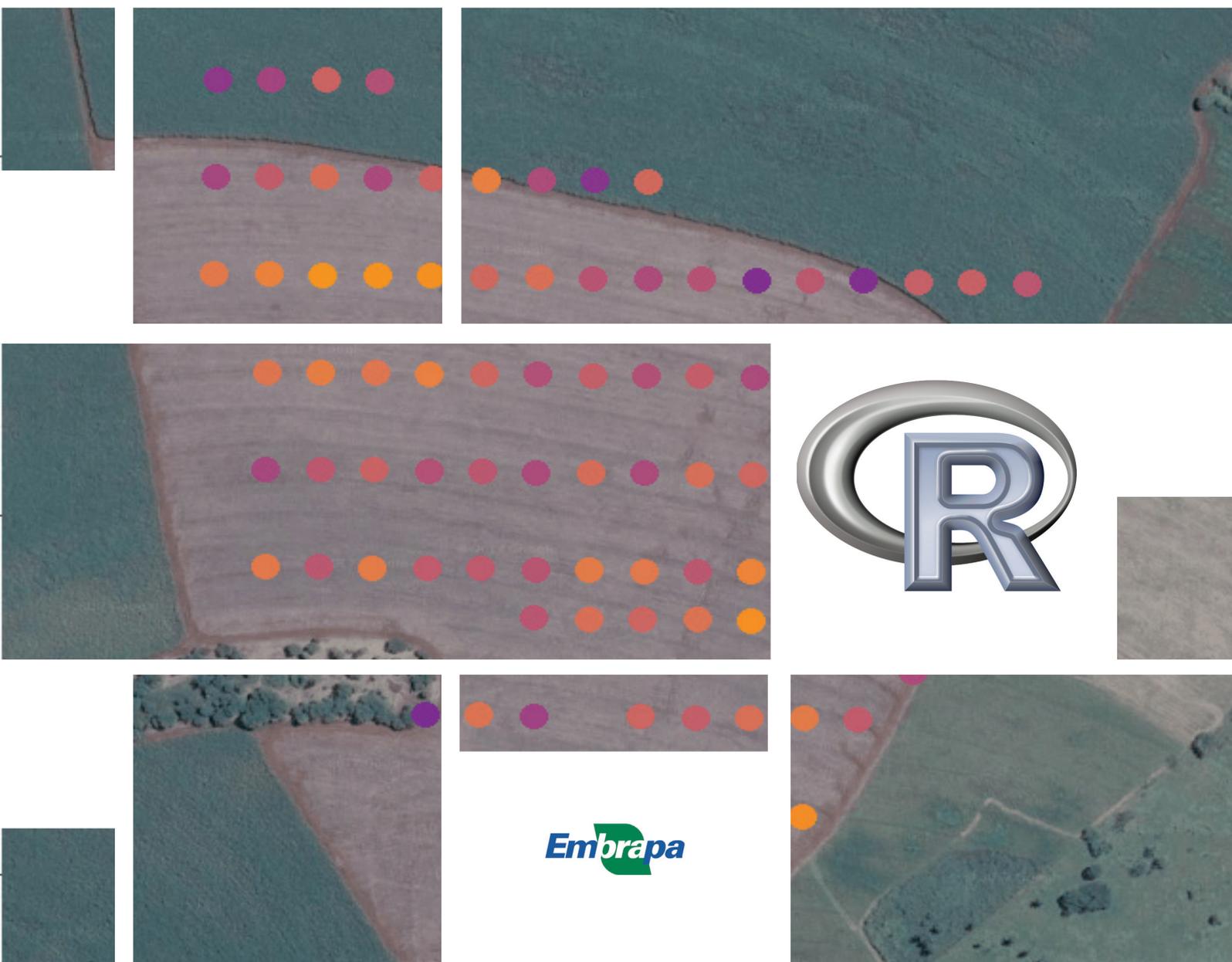


Visualização de dados espacializados usando o R



*Empresa Brasileira de Pesquisa Agropecuária
Embrapa Monitoramento por Satélite
Ministério da Agricultura, Pecuária e Abastecimento*

DOCUMENTOS 115

Visualização de dados espacializados usando o R

Fernando Antônio de Pádua Paim

Exemplares desta publicação podem ser adquiridos na:

Embrapa Monitoramento por Satélite

Av. Soldado Passarinho, nº 303
Fazenda Jardim Chapadão
13070-115, Campinas, SP
Fone: (19) 3211.6200
www.embrapa.br/territorial
www.embrapa.br/fale-conosco/sac

Comitê Local de Publicações
da Embrapa Monitoramento por Satélite

Presidente
Sérgio Gomes Tosto

Secretário-Executivo
Bibiana Teixeira de Almeida

Membros
André Luiz dos Santos Furtado, Bibiana Teixeira de Almeida, Carlos Fernando Quartaroli, Daniela Maciel Pinto, Fabio Enrique Torresan, Gustavo Bayma Siqueira da Silva, Janice Freitas Leivas, Marcelo Fernando Fonseca, Vera Viana dos Santos Brandão

Supervisão editorial
Suzi Carneiro e Bibiana Teixeira de Almeida

Revisão de texto
Bibiana Teixeira de Almeida

Normalização bibliográfica
Vera Viana dos Santos Brandão

Projeto gráfico da coleção
Carlos Eduardo Felice Barbeiro

Editoração eletrônica e Tratamento das ilustrações
Suzi Carneiro

Ilustração da capa
Suzi Carneiro

1ª edição
1ª impressão (2017): versão on-line

Todos os direitos reservados.

A reprodução não autorizada desta publicação, no todo ou em parte, constitui violação dos direitos autorais (Lei nº 9.610).

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Monitoramento por Satélite

Paim, Fernando Antônio de Pádua.

Visualização de dados espacializados usando o R / Fernando Antônio de Pádua
Paim. – Campinas : Embrapa Monitoramento por Satélite, 2017.
20 p. : il. ; (Documentos / Embrapa Monitoramento por Satélite,
ISSN 0103-7811; 115).

1. Informação georreferenciada. 2. Programa de computador. 3. Sistema de
informação geográfica. I. Título. II. Série.

CDD 006

© Embrapa, 2017

Autor

Fernando Antônio de Pádua Paim

Analista de Sistemas, Especialista em Análise de Sistemas, analista da Embrapa Monitoramento por Satélite, Campinas, SP

Sumário

Apresentação	9
Introdução.....	10
O experimento canamogi	10
Ajuste do ambiente do R.....	11
Carga das bibliotecas necessárias.....	11
Os dados do experimento	12
Leitura de dados de um arquivo de texto	12
A função get_map.....	14
Conclusão.....	19
Agradecimentos.....	20
Referência consultada.....	20

Apresentação

A visualização dos dados de um campo experimental sobre a imagem real de uma região de estudo é um recurso interessante que as geotecnologias atuais proporcionam. Dessa forma, obtém-se um mapa que facilita a observação, análise e tomada de decisão. O software R, por meio do pacote `ggmap`, disponibiliza funcionalidades que possibilitam combinar um conjunto de dados com coordenadas geográficas e uma imagem de fundo real tomada nas mesmas coordenadas. Este tutorial sucinto descreve uma sugestão dos passos para promover esse ajuntamento. O documento não descreve a resolução de um problema específico, mas trata de um exemplo genérico que explica de modo não exaustivo uma das formas de fazer a espacialização de dados usando o R. O texto contém uma breve descrição dos comandos, mas é desejável que o usuário tenha alguma familiaridade com o ambiente do R para executar esses comandos.

Introdução

Uma característica essencial de um experimento geoestatístico é a presença de coordenadas geográficas associadas aos pontos amostrados. Segundo VIEIRA (2000), o semivariograma, entidade matemática que expressa a autocorrelação dos valores da variável em estudo, dado pela Equação 1, indica que a semivariância, $\gamma(h)$, é função apenas da distância h (vetor, em metros) entre os pares de pontos considerados.

$$\gamma(h) = \frac{1}{2n_h} \sum_{i=1}^{n_h} [z_{(i+h)} - z_{(i)}]^2 \quad (1)$$

Assim, se o objetivo é apenas calcular as semivariâncias e plotá-las em um gráfico de dispersão em função da distância h , não há necessidade de atenção maior com relação ao sistema de projeção das coordenadas que, naquele cálculo, são utilizadas expressas em Universal Transversa de Mercator (UTM). No entanto, para representar os dados do experimento sobre uma imagem real da área estudada, é necessário levar em consideração o sistema de coordenadas utilizado.

O presente tutorial mostra uma forma de utilizar funcionalidades do R para visualizar dados espacializados segundo um sistema de coordenadas, o que implica definir o sistema de projeção a ser utilizado.

O experimento canamogi

Os dados utilizados neste tutorial foram obtidos do experimento descrito em GREGO et al. (2014):

O objetivo do trabalho foi a integração de informações espaciais de solo e planta para identificação das zonas homogêneas na Unidade Piloto com cana-de-açúcar, Mogi Mirim, SP. Os dados foram coletados na área com cana-de-açúcar na safra 2011/2012 com grade mínima de 90 pontos. Foram coletadas amostras de solo e planta: granulometria (15 cm); densidade do solo (15 e 30 cm); resistência à penetração do solo (0–10, 10–20 e 20–30 cm); número de colmos (colmos m-2); índice de vegetação (NDVI) pelo sensor Crop Circle; condutividade elétrica do solo pelo sensor Veris (30 e 90 cm).

Para este trabalho, os dados do experimento canamogi considerados dizem respeito apenas à produtividade da cana-de-açúcar, em toneladas por hectare, e foram tomados em 111 pontos georreferenciados em uma área de aproximadamente 17 ha. O campo experimental, aqui denominado simplesmente de canamogi, está localizado em Mogi Mirim, SP. A atividade, desenvolvida junto com parceiros da rede de agricultura de precisão da Embrapa, consistiu na coleta dos dados com identificação dos pontos usando equipamento de GPS da Garmin. A metodologia da pesagem consistiu na contagem de colmos de cana em linhas de 10 m. Ainda segundo GREGO et al. (2014):

De acordo com os resultados houve dependência espacial para a maioria dos atributos do solo e da planta, e a interpolação permitiu visualizar a divisão da área em duas partes, uma ao norte da área e outra ao sul, separadas pela pedreira identificada na imagem de satélite e na imagem do veículo aéreo não tripulado. Também foram identificadas duas regiões distintas quanto ao número de colmos e o índice de vegetação. A condutividade elé-

trica apresentou dependência espacial de fraca a nula e não foi identificada relação espacial com os demais atributos.

Ajuste do ambiente do R

Ao iniciar uma sessão de trabalho com o R, é boa prática ajustar o ambiente da ferramenta para o uso dos comandos, a análise dos resultados e a gravação dos dados. O ajuste do ambiente geralmente consiste em liberar a memória de objetos que não serão utilizados, definir o diretório ou pasta de trabalho e carregar as bibliotecas ou pacotes que serão utilizados.

A função `ls` lista todos os objetos que estão atualmente armazenados na memória de trabalho do R e que foram carregados a partir da sessão anterior. A função `rm` faz a remoção dos objetos da memória e libera a memória para uso. A chamada `rm(list=ls())` é bem usual e remove todos os objetos da memória. A função `getwd` exibe o *path* (nome e caminho) do diretório de trabalho atualmente selecionado. Para alterar o diretório de trabalho, é utilizada a função `setwd`. Abaixo é mostrado o código R utilizado para fazer esses ajustes iniciais. A pasta (ou diretório) de trabalho pode ser qualquer uma. No exemplo foi utilizada a pasta `C:\Fernando\R\canamogi ggmap\`.

```
rm(list=ls())
> ls()
character(0)
> setwd("C:/Fernando/R/canamogi ggmap")
> getwd()
[1] "C:/Fernando/R/canamogi ggmap"
```

Mais informações estão disponíveis na ajuda do ambiente, por meio dos comandos `help(ls)`, `help(rm)`, `help(getwd)`, `help(setwd)`.

Carga das bibliotecas necessárias

As bibliotecas (ou pacotes) `ggmap` e `sp` são os recursos utilizados neste trabalho, para criar a visualização de dados espacializados. Com o pacote `ggmap` é baixada a imagem de satélite que será utilizada como imagem de fundo, sobre a qual são representados os pontos amostrados. O pacote `sp` é utilizado para definir o sistema de projeção das coordenadas. O pacote `ggmap` necessita do pacote `ggplot2`.

As instruções abaixo fazem o R carregar os pacotes para o seu ambiente. Caso o pacote não esteja instalado, use o comando `install.packages("nome-do-pacote", dependencies=TRUE)` para proceder à instalação, onde `nome-do-pacote` deve ser substituído por `ggplot2`, `ggmap` ou `sp`. Mais informações estão disponíveis na ajuda do ambiente, por meio do comando `help(install.packages)` no *prompt* do R.

```
library(ggplot2)
library(ggmap)
library(sp)
```

O exemplo abaixo, no qual são recuperadas as informações de latitude e longitude da cidade de Brasília, exemplifica a praticidade e funcionalidade dessas bibliotecas.

```
geocode(location="brasilia")
```

```
Information from URL : http://maps.googleapis.com/maps/api/geocode/  
json?address=brasilia&sensor=false
```

```
lon lat  
1 -47.88253 -15.79416
```

A função `geocode` utilizada no exemplo acima é um exemplo das funcionalidades disponíveis no pacote `ggmap`. Não é objetivo deste documento descrever as funções disponíveis nos pacotes utilizados ou fazer uma descrição detalhada das funções utilizadas, mas apenas ilustrar uma forma de usá-las. Mais informações estão disponíveis na ajuda do ambiente, por meio do comando `help(-geocode)` no *prompt* do R.

Os dados do experimento

Os dados do experimento (as coordenadas geográficas e a produtividade da cana-de-açúcar) estão disponíveis no arquivo `faz_aparecida.csv`, o qual pode ser baixado diretamente da nuvem da Embrapa no endereço <https://cloud.sede.embrapa.br/owncloud/s/948HeQwomS43FYh>. Também é possível baixar o arquivo de dentro do ambiente do R usando o comando:

```
https://cloud.sede.embrapa.br/owncloud/s/948HeQwomS43FYh/download
```

```
download.file(url="https://cloud.sede.embrapa.br/owncloud/s/948HeQwomS43FYh", destfile="faz_aparecida.csv").
```

O arquivo ainda pode ser obtido do repositório da Embrapa Monitoramento por Satélite usando o comando:

```
download.file(url="http://repositorio.cnpm.embrapa.br/ftp/fernando/faz_aparecida.csv", destfile="faz_aparecida.csv")
```

Leitura de dados de um arquivo de texto

Um arquivo no formato Comma-Separated Values (CSV, valores separados por vírgulas) é um arquivo American Standard Code for Information Interchange (ASCII, código padrão americano para o intercâmbio de informação) puro. No caso de um arquivo CSV com os resultados de um experimento, os dados apresentam-se dispostos em linhas com um caractere separador para as variáveis do estudo (as colunas). O caractere separador pode ser vírgula, ponto-e-vírgula, tabulação, etc. A primeira linha do arquivo pode ou não conter os nomes das colunas dos dados. Um arquivo CSV pode ser gerado diretamente num editor de texto ASCII – por exemplo, o Bloco de Notas do Windows – com as anotações das variáveis do experimento, obedecendo um protocolo definido pelo próprio autor das anotações. Os dados podem ser gerados a partir de algum instrumento que procede à sua captação e posteriormente são descarregados, na forma de arquivo de texto, em um microcomputador, para serem processados. É necessário observar o protocolo utilizado pelo instrumento para a geração dos dados. Geralmente os dados são apresentados em uma planilha do Libre Office Calc ou Microsoft Excel, e essas ferramentas dispõem da funcionalidade de exportar ou salvar os dados em formato CSV.

O comando abaixo é usado para abrir um arquivo CSV e carregar o seu conteúdo para o ambiente do R.

```
cana = read.table(
  file="C:/Fernando/R/canamogi_ggmap/faz_aparecida.csv",
  header=TRUE,
  sep=';')
```

```
names(cana)
```

```
[1] "x" "y" "tph"
```

```
>
```

A chamada abaixo cria uma variável de nome `cana` do tipo *dataframe*. *Dataframe* é o tipo gerado pelo R toda vez que é usada a função `read.table`. O comando especifica que o arquivo com os dados a ser lido, especificado no parâmetro `file`, é o de nome `faz_aparecida.csv`. A primeira linha do arquivo contém os nomes das colunas, e essa informação é passada ao R no parâmetro `header=TRUE`. O caractere separador de colunas no arquivo é o ponto-e-vírgula, e essa informação é passada no parâmetro `sep=";"`. A função `class` aplicada à variável `cana`, como em `class(cana)`, exibe o tipo de objeto da variável, no caso, *dataframe*.

```
cana = read.table(
  file="C:/Fernando/R/canamogi_ggmap/faz_aparecida.csv",
  header=TRUE,
  sep=';')
```

Os nomes das variáveis ou das colunas do *dataframe* são exibidos usando a função `names` e tendo como argumento o nome do *dataframe*, como em `names(cana)`. O caractere cifrão (\$) seguido do identificador do campo, sem espaços entre eles, é a forma de fazer referência individualizada aos campos internos (variáveis ou colunas) de um *dataframe*. Assim, `cana$x` faz referência à coluna de nome `x` do *dataframe* `cana`.

A função `head` exibe dados do início do *dataframe*, e a função `tail` exibe dados do fim. Mais informações estão disponíveis na ajuda do ambiente, por meio dos comandos `help(head)` e `help(tail)` no *prompt* do R.

```
> head(cana)
```

```
      x      y    tph
1 299621 7505136  91.89
2 299621 7505196 151.41
3 299621 7505256 168.36
4 299651 7504956  91.29
5 299651 7505016 150.09
6 299651 7505076  95.24
```

```
> tail(cana)
```

```
      x      y    tph
106 300041 7505136 119.23
107 300071 7505016 127.29
108 300071 7505046 108.53
```

```
109 300071 7505076 187.28
110 300071 7505136 132.13
111 300101 7505076 161.49
```

Um gráfico das coordenadas x e y e um gráfico dos valores proporcionais da variável de estudo tph (toneladas por hectare) são mostrados nas Figuras 1 e 2.

```
plot(cana$x, cana$y, asp=1, main="Campo experimental")
```

```
plot(cana$x, cana$y, asp=1,
     main="Campo experimental",
     cex=cana$tph/max(cana$tph),
     pch=19)
```

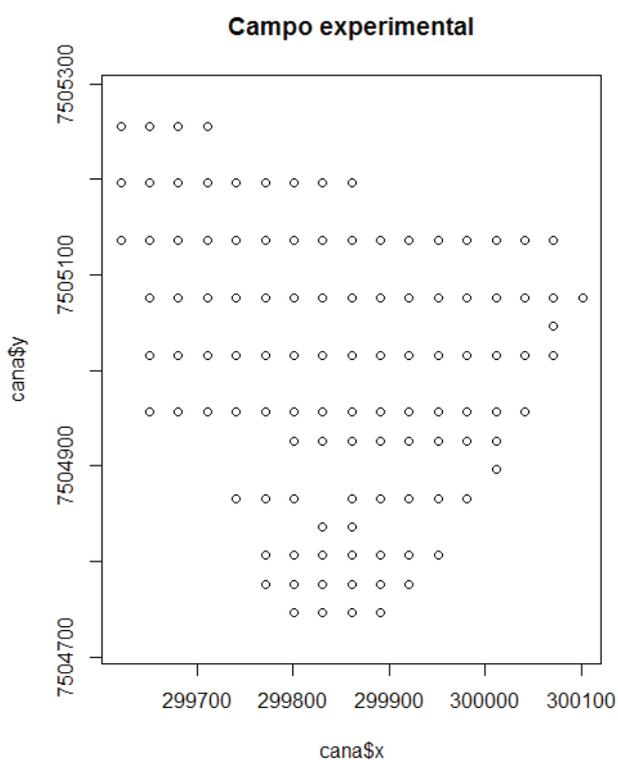


Figura 1. Gráfico com a representação do campo experimental nas coordenadas UTM.

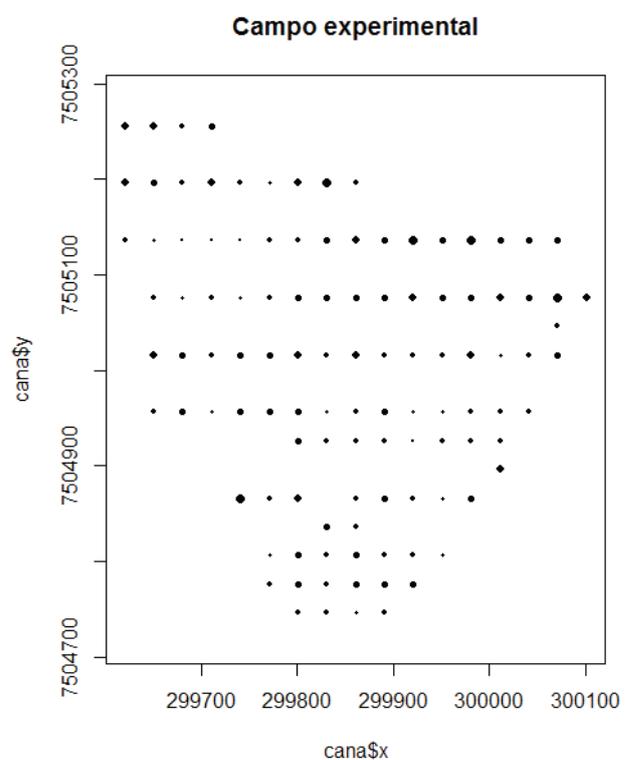


Figura 2. Gráfico com a representação do campo experimental com os valores relativos da variável de estudo em função do maior valor $\max(\text{cana}\$tph)$.

A função `get_map`

A função `get_map` do pacote `ggmap` tem a seguinte sintaxe básica:

```
get_map(location, maptype, source, crop, zoom)
```

Ela consulta os servidores do Google Maps, OpenStreetMap, Stamen Maps ou Naver Map e recupera, se existir, um mapa de acordo com os parâmetros informados.

O parâmetro `location` deve ser expresso em longitude e latitude, o parâmetro `maptype` é um valor do tipo *character string* e define o tema desejado para o mapa, `source` define o servidor no

qual será feita a consulta, o parâmetro `crop`, do tipo lógico, especifica se a imagem será delimitada e, `zoom`, um inteiro no intervalo [3,21], define o nível de detalhamento do mapa.

E necessário especificar o sistema de projeção das coordenadas dos pontos amostrados, para recuperar a imagem adequada. As coordenadas, especificadas nas colunas `x` e `y` no *dataframe* `cana` estão expressas em UTM e usam o *datum* WGS84 segundo GREGO et al. (2014).

Nota: A função `get_map` espera receber as coordenadas no parâmetro `location` na forma de longitudes e latitudes, portanto uma conversão de coordenadas deve ser feita.

A função `proj4string` faz a atribuição de um sistema de projeção de coordenadas a um conjunto de coordenadas. Há um padrão na forma de especificar o sistema a ser utilizado. Mais informações estão disponíveis por meio do comando `help(proj4string)`.

Os passos a serem seguidos são:

- 1) atribuir um sistema de projeção aos dados originais do *dataframe* `cana`,
- 2) fazer a conversão das coordenadas para o sistema de longitudes e latitudes,
- 3) criar um segundo *dataframe* com os novos dados e criar o mapa.

A sequência de comandos abaixo é utilizada na execução dos passos acima:

```
c1 = "+proj=utm +zone=23 +south +datum=WGS84 +units=m +no_defs"
c2 = "+proj=longlat +zone=23 +south +datum=WGS84 +units=m +no_defs"
c3 = "+proj=utm +datum=WGS84"

xy1 = data.frame(x=cana$x, y=cana$y)
coordinates(xy1) = c('x', 'y')
proj4string(xy1) = CRS(c1)
xy2 = spTransform(xy1, CRS(c2))
cana2 = data.frame(x=xy2$x, y=xy2$y, z=cana$tp)
```

Foi preservado o *dataframe* original, `cana`, e foi criado o *dataframe* `cana2` com os mesmos dados. Após a chamada da função `coordinates(xy1)=c('x', 'y')`, a classe do objeto `xy1`, obtida usando `class(xy1)`, que anteriormente era *dataframe*, passou a ser `SpatialPointsDataFrame`. Dessa forma, foi possível atribuir um sistema de projeção às coordenadas. Isso é feito usando a função `proj4string` na linha `proj4string(xy1)=CRS(c1)`. A função `Coordinate Reference System (CRS)` atribui um sistema de referência a um objeto. Mais informações estão disponíveis por meio do comando `help(CRS)` no *prompt* do R.

Uma listagem do objeto `xy1` exhibe os valores mostrados abaixo.

```
> head(xy1)
SpatialPoints:
           x           y
[1,] 299621 7505136
[2,] 299621 7505196
[3,] 299621 7505256
[4,] 299651 7504956
[5,] 299651 7505016
[6,] 299651 7505076
```

Coordinate Reference System (CRS) arguments: `+proj=utm +zone=23 +south +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0 >`

As coordenadas aparecem sob o nome `SpatialPoints` e é possível fazer referência individualizada às coordenadas `x` e `y`, com `xy1$x` e `xy1$y`.

Foram plotados gráficos do campo experimental utilizando `plot(xy1$x, xy1$y, asp=1, main="Campo experimental")` ou `plot(xy1, asp=1, main="Campo experimental")`. As Figuras 3 e 4 mostram a saída gráfica.

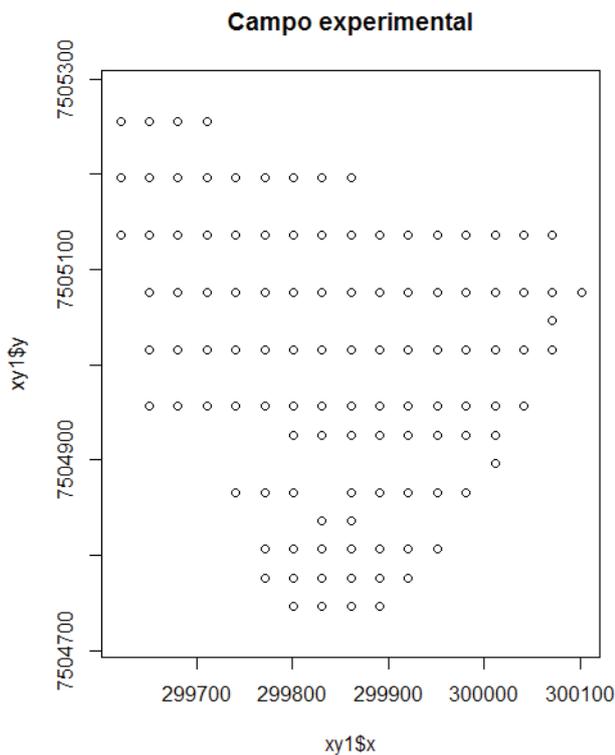


Figura 3. Gráfico com a representação do campo experimental dado pelos novos pontos `xy1$x` e `xy1$y` com especificação da projeção.

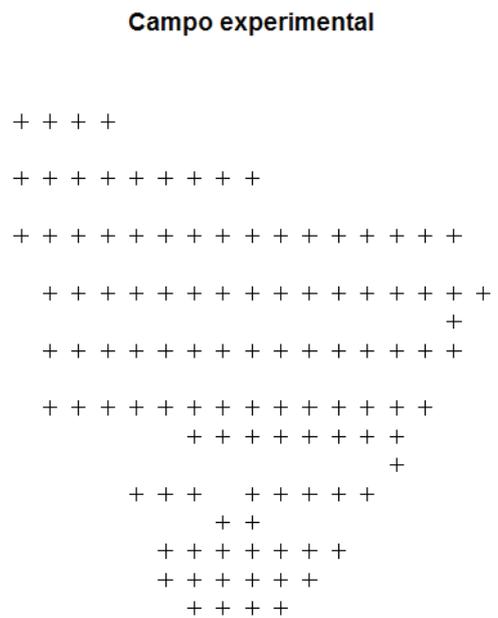


Figura 4. Gráfico com a representação do campo experimental com os pontos especificados em projeção `plot(xy1)`.

A conversão de coordenadas para o sistema longitude/latitude é feita por meio da função `spTransform`, da forma: `xy2=spTransform(xy1, CRS(c2))`. O novo `dataframe`, `cana2`, é criado usando `cana2=data.frame(x=xy2$x, y=xy2$y, z=cana$tp)`.

Uma listagem do objeto `xy2` exhibe os valores das coordenadas no novo sistema de referência. As coordenadas aparecem sob a forma de longitude e latitude com o nome `SpatialPoints`, e `xy2$x` e `xy2$y` são referências individualizadas para as coordenadas `x` e `y`, respectivamente.

```
> head(xy2)
SpatialPoints:
          x          y
[1,] -46.94858 -22.54865
[2,] -46.94858 -22.54811
[3,] -46.94857 -22.54757
[4,] -46.94831 -22.55028
```

```
[5,] -46.94831 -22.54974
[6,] -46.94830 -22.54920
Coordinate Reference System (CRS) arguments: +proj=longlat
+datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0 >
```

Foram plotados gráficos do campo experimental utilizando `plot(xy2$x, xy2$y, asp=1, main="- Campo experimental")` e `plot(xy2, asp=1, main="Campo experimental")`.

Os resultados são mostrados nas Figuras 5 e 6.

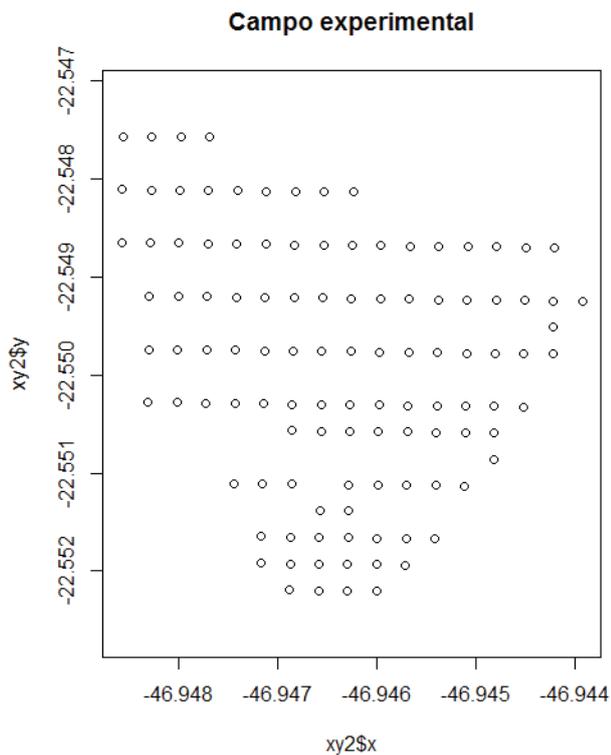


Figura 5. Gráfico com a representação do campo experimental com os pontos especificados em longitude e latitude, plotado usando `plot(xy2$x,xy2$y)`.

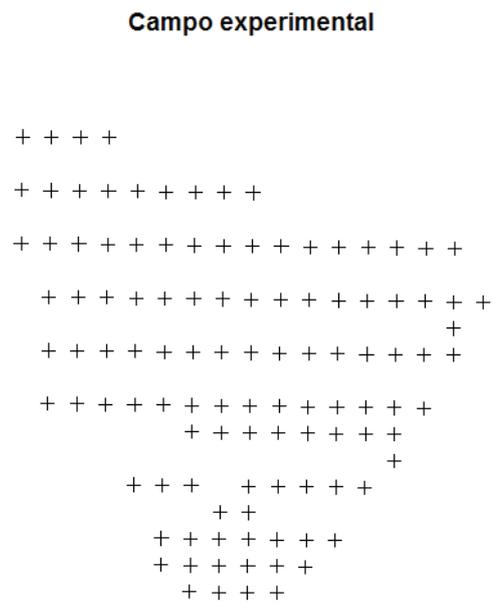


Figura 6. Gráfico com a representação do campo experimental com os pontos especificados em longitude e latitude, plotado usando `plot(xy2)`.

Uma vez criado o novo *dataframe*, `cana2`, com as coordenadas expressas em longitudes e latitudes, é possível recuperar uma imagem da área de interesse usando a função `get_map`.

O código R abaixo exemplifica a utilização da função `get_map`. No exemplo foi especificado o valor da mediana (*median*) das coordenadas como parâmetros de referência para a longitude e a latitude, pois a função `get_map` recupera a imagem contida no retângulo envolvente a partir dos valores de referência.

```
canamap = get_map(
  location = c(lon=median(cana2$x), lat=median(cana2$y)),
  matype = "satellite",
  source = "google",
```

```
crop = FALSE,  
zoom = 17)
```

Após a execução da função `get_map`, o ambiente do R retorna com texto similar a este:

```
Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=-22.54977,-46.9463&zoom=17&size=640x640&scale=2&maptype=satellite&language=en-EN&sensor=false
```

Uma forma de trabalhar com a função `ggmap` consiste em adicionar recursos cumulativamente até que, por fim, seja feita a plotagem do gráfico. No código abaixo foi definida a origem dos dados no objeto `cana2`, foi adicionada uma legenda e foi definido um gradiente de cores para destacar os valores de `z`. Em seguida, o mapa foi plotado.

```
canaplot = ggmap(canamap) +  
  geom_point(  
    aes_string(x="x", y="y", color=cana2$z),  
    size = 5,  
    data = cana2)  
  
canaplot = canaplot +  
  theme(legend.position = "bottom")  
  
canaplot = canaplot +  
  scale_color_gradient(low="darkorange", high="darkorchid4")  
  
plot(canaplot)
```

A função `theme` define vários parâmetros do gráfico. O parâmetro `legend.position` define a posição da legenda. No exemplo, a legenda é exibida na parte inferior (*bottom*).

A função `scale_color_gradiente` define um gradiente de cores a partir de duas cores que estabelecem o menor valor e o maior valor (Figura 7).

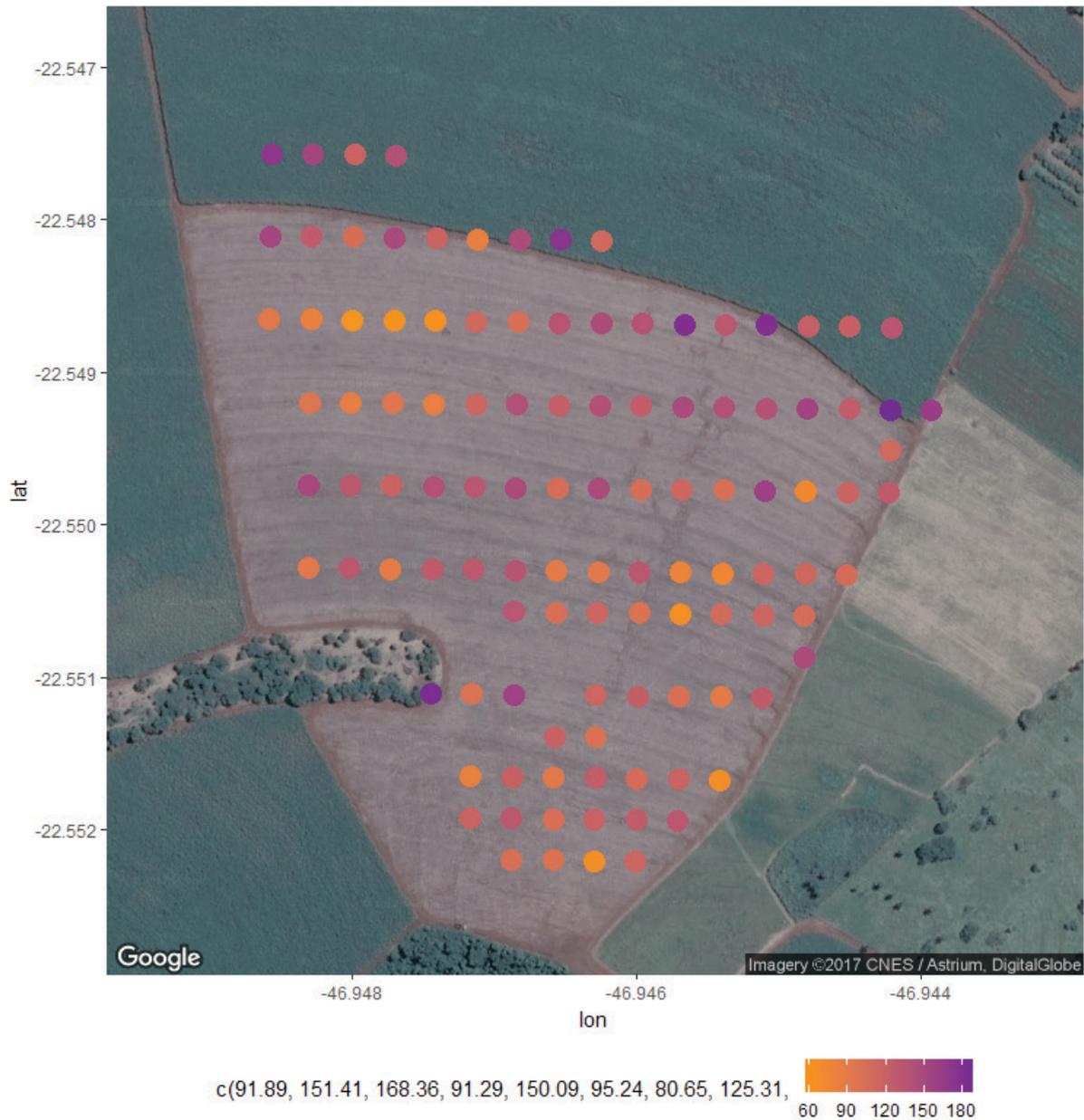


Figura 7. Gráfico com a representação do campo experimental com os pontos especificados em longitude e latitude sobre a área real do experimento na Fazenda Aparecida, no Município de Mogi Mirim, São Paulo.

Nota: Há um deslocamento entre os pontos amostrados e a imagem real do campo. É possível que a aplicação de um fator de correção do GPS utilizado ajuste as posições, mas essa possibilidade não foi verificada neste trabalho.

Conclusão

Confeccionar mapas por meio de uma linguagem de programação é um recurso bastante poderoso. De um lado, possibilita a visualização dos dados na forma de mapas e, de outro, agrega toda uma funcionalidade matemática e estatística disponível para as mais diversas operações. O objetivo deste trabalho foi o de agregar valor ao experimento canamogi e, ao mesmo tempo, mostrar como o R pode auxiliar nessas tarefas. Com apenas duas funções básicas, `get_map` e `ggmap`, foi possível disponibilizar resultados de um experimento sobre a área real onde ele aconteceu.

Agradecimentos

À equipe do projeto em rede de Agricultura de Precisão da Embrapa "Inovação tecnológica em Agricultura de Precisão - PC5 (Inova_AP)" (Código SEG 01.09.01.002.05.00), pela aquisição dos dados na unidade piloto de cana-de-açúcar da Fazenda Aparecida em Mogi Mirim, SP, e pela autorização do uso desses dados nesta obra. Ao Dr. Sidney, por disponibilizar uma cópia do documento "Geoestatística em estudos de variabilidade espacial do solo" (VIEIRA, 2000) e pelas constantes trocas de ideias sobre Geoestatística.

Referência consultada

GREGO, C. R.; ARAUJO, L. S.; VICENTE, L. E.; NOGUEIRA, S. F.; MAGALHAES, P. S. G.; VICENTE, A. K.; BRANCALIÃO, S. R.; VICTORIA, D. C.; BOLFE, E. L. Agricultura de precisão em cana-de-açúcar. In: Bernardi, Alberto Carlos de Campos; Naime, João de Mendonça; Resende, Álvaro Vilela de.; Bassoi, Luis Henrique; Inamasu, Ricardo Yassushi.. (Org.). Agricultura de precisão: resultados de um novo olhar. 2ed. Brasília, DF: Embrapa, 2014, v. 1, p. 442-457.

Ajuda interna do R acessada localmente (na própria máquina) com o navegador Firefox.

<http://127.0.0.1:18292/library/base/html/cbind.html>

<http://127.0.0.1:18292/library/base/html/data.frame.html>

http://127.0.0.1:18292/library/ggmap/html/get_map.html

http://127.0.0.1:18292/library/ggplot2/html/guide_legend.html

<http://127.0.0.1:18292/library/ggplot2/html/guides.html>

<http://127.0.0.1:18292/library/graphics/html/plot.html>

<http://127.0.0.1:28700/library/ggmap/html/ggmap.html>

http://127.0.0.1:28700/library/ggplot2/html/scale_gradient.html

<http://127.0.0.1:28700/library/ggplot2/html/theme.html>

<http://127.0.0.1:28700/library/sp/html/CRS-class.html>

<http://127.0.0.1:28700/library/sp/html/is.projected.html>

<http://127.0.0.1:28700/library/utils/html/download.file.html>

R Core Team (2016). **R**: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

VIEIRA, S. R. Geoestatística em estudos de variabilidade espacial do solo. In: NOVAIS, R. F.; ALVAREZ, V. H.; SCHAEFER, G. R. (Ed.). Tópicos em Ciência do solo. Viçosa: Sociedade Brasileira de Ciência do Solo, 2000. p. 1-54. v. 1. VIEIRA, S. R.; MILLETE, J; TOPP, G. C.; REYNOLDS



Monitoramento por Satélite