

# Visualização Espacial De Dados Públicos

Elias Teixeira Krainski

Novembro, 2011

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Dados Públicos</b>	<b>3</b>
2.1	Dados Públicos e Publicação de Dados . . . . .	3
2.2	Algumas Fontes de Dados . . . . .	4
2.3	Obtenção dos Dados . . . . .	5
2.3.1	Dados do IBGE . . . . .	5
2.3.2	Dados do Datasus . . . . .	6
2.4	Importação em <b>R</b> . . . . .	7
2.4.1	Arquivos CSV . . . . .	7
2.4.2	Arquivos DBC . . . . .	8
2.5	Exercícios . . . . .	9
<b>3</b>	<b>Mapas</b>	<b>11</b>
3.1	Fonte de Mapas Territoriais no Brasil . . . . .	11
3.2	Importando um <i>shapefiles</i> em <b>R</b> . . . . .	12
3.3	Georeferenciamento Espacialmente Discreto . . . . .	13
3.4	Mapa temático . . . . .	14
3.5	Exercícios . . . . .	17
<b>4</b>	<b>Georeferenciamento</b>	<b>19</b>
4.1	Georeferenciamento de Logradouro . . . . .	19
4.2	Localização de Endereço num Setor Censitário . . . . .	21
4.2.1	Mapa de Setores Censitários . . . . .	21
4.2.2	Projeção Cartográfica . . . . .	22
4.2.3	Endereço Pontual em Setor Censitários . . . . .	23
4.3	Georeferenciamento de CEP . . . . .	24
4.4	Imagens do GoogleMaps . . . . .	26
4.4.1	Obtendo Imagens do GoogleMaps . . . . .	26
4.4.2	Georeferenciamento de imagens . . . . .	27

# Capítulo 1

## Introdução

Neste material nós apresentamos alguns recursos disponíveis em ambiente **R** [?] para a visualização de dados georeferenciados. Vamos abordar os diferentes tipos de dados disponíveis, as principais fontes de dados, as formas de importar dados em **R**, o georeferenciamento e a visualização. A abordagem será de forma prática, ilustrada com diversos exemplos.

Muitos órgãos públicos e/ou pesquisa disponibilizam dados de forma sistemática. Muitos desses dados possuem algum tipo de referência geográfica. Esses dados podem ser as informações demográficas, índices socio-econômicos, dados de saúde, dados de clima, entre outros. Eles podem representar dados de unidades administrativas políticas (países, estados, municípios, etc.), de uma área definida arbitrariamente, de endereços postais ou de um ponto geográfico.

Um dado com referência espacial (ou geográfica) nada mais é um dado que contém a informação do local a que se refere, ou que foi coletado. Assim como um dado temporal contém a data de quando se refere ou foi coletado. Por exemplo, se tivermos duas colunas numa tabela, uma indicando cada estado brasileiro e outra com a taxa de analfabetismo de cada estado, teremos então essa taxa referenciada por estado. A referência geográfica fica completa ao se obter as coordenadas dos polígonos que representam cada estado. Quando temos por exemplo a ocorrência de um crime, geralmente temos também o endereço (logradouro) onde a ocorrência se passou. Neste caso precisamos obter a coordenada geográfica desse endereço, isto é, fazer o georeferenciamento do dado. Outro exemplo é quando temos dados de estações meteorológicas (temperatura, precipitação, etc.) e temos a coordenada geográfica de cada estação.

Inicialmente nós vamos apresentar algumas fontes de dados, o formato digital dos dados e como importá-los no ambiente **R**. Muitos dos dados apresentados aqui serão de municípios brasileiros ou setores censitários. Neste caso vamos precisar também de mapas digitais da malha de municípios ou setores censitários. Neste caso vamos considerar a malha digital disponibilizada pelo Instituto Brasileiro de Geografia e Estatística - IBGE.

Alguns dados de saúde estão referenciados por Código de Endereçamento Postal - CEP. Neste caso, nós vamos usar uma base de dados de CEPs, disponível na Internet, para obter o logradouro. Com o logradouro, podemos usar uma API (*Application*

*Program Interface*) de georeferenciamento do GoogleMaps. Esta API permite a obtenção das coordenadas geográficas (Latitude e Longitude) a partir de um endereço dado por logradouro.

Nós geralmente denominamos dados à informações. Quando esses dados estão georeferenciados, nós os chamamos de dados espaciais. O processo de georeferenciar e visualizar dados espaciais é também conhecido como geoprocessamento. O geoprocessamento comumente é feito num Sistema de Informações geográficas (SIG).

Um SIG é um *software* especialista em operações espaciais, tais como projeções, cálculo de distâncias, de estruturas de vizinhança, consultas espaciais, etc. Um SIG geralmente implementa métodos estatísticos simples e alguns deles implementam métodos simples de Estatística Espacial. Mas neste último quesito, nenhum deles é completo.

O Instituto Nacional de Pesquisas Espaciais tem um SIG que é gratuito, o SPRING [?], e um que é gratuito e de código aberto (*open source*), o TerraView [?], que é baseado na, também *open source*, TerraLib [?].

O **R** é uma linguagem e ambiente voltado para análises estatísticas. Existem vários pacotes adicionais em **R** que tornam possíveis todas as operações disponíveis em um SIG. Assim, podemos fazer georeferenciamento, visualização, operações espaciais e, principalmente, análise estatística de dados espaciais usando o **R**. O georeferenciamento, porém, é feito usando uma API de georeferenciamento do GoogleMaps. Nós apenas fazemos consulta a essa API a partir do ambiente **R**. Outra possibilidade é a busca de imagens (de ruas, de satélite, etc.) na API de visualização do GoogleMaps, nos possibilitando sobrepor uma imagem a um mapa e vice-versa.

Neste material, nós apenas vamos apenas mostrar algumas das funcionalidades úteis para a análise de dados espaciais, usando formatos tradicionais de mapas e imagens, (*shapefiles* e arquivos de imagens). Porém, recentemente tem sido desenvolvidos em **R** também funcionalidades para se trabalhar com o formato kml. Esperamos que este seja um material de ajuda aos iniciantes na Estatística Espacial.

# Capítulo 2

## Dados Públicos

Não pretendemos listar exaustivamente as fontes de dados públicos, muito provavelmente o número de fontes não listadas neste material excede o de fontes listadas. Nós vamos considerar fontes de dados demográficos e socio-econômicos, dados de saúde e dados de clima. Há, por exemplo, órgãos que disponibilizam outros tipos de dados, tais como de educação, de crimes, etc.

Inicialmente, vamos classificar os dados em três grupos: os microdados, dados tabulados a partir de microdados e dados observacionais. Microdados são os dados de cada questionário de uma pesquisa. Suponha por exemplo que fazemos uma entrevista com várias pessoas, os microdados são as respostas de cada pessoa. Dados tabulados de microdados são uma agregação dos dados a um nível mais alto. Por exemplo, a média de idade das pessoas da pesquisa por bairro. Os dados observacionais, por sua vez, são dados coletados de observações de um fenômeno. Por exemplo, observamos a precipitação acumulada por dia num determinado local.

### 2.1 Dados Públicos e Publicação de Dados

Diversos órgãos públicos brasileiros tem tido uma política de abertura no sentido de disponibilizar dados. Talvez isso tenha ocorrido devido à força do argumento de que dados coletados com recursos públicos também deveriam ser de domínio público, ou seja, dados públicos deveriam ser publicados. Mas muito dessa abertura é devido ao avanço dos recursos computacionais e tecnológicos. A importância de se ter dados é que se pode utilizá-los em pesquisas e para subsídio políticas públicas. Infelizmente há ainda muitos dados que seriam úteis em diversas pesquisas que não são públicos. Alguns, tais como os de saúde suplementar, telefonia, etc., não são públicos, por motivos óbvios.

Alguns argumentam que disponibilizar microdados é anti-ético, pela sempre presente possibilidade de identificação da fonte primária dos dados. Há também argumentos no sentido de que o mau uso de dados pode ser perigoso. Por exemplo, imagine que numa pesquisa do IBGE seja possível identificar algumas residências de um setor censitário, apenas observando as respostas tais como número de pessoas e característica (etária, etc.) de cada uma.

Mas o que pensar quando artigos que discutem saúde pública, num nível que seria difícil, ou até impossível, sem a publicação desses dados? Por exemplo, considere que está disponível informações de cada recém nascido. Isto é, são disponibilizados dados demográficos, socio-econômicos da mãe do nascido vivo, dados de assistência à saúde durante a gestação, dados do parto e das condições do nascido vivo. Ou seja, qualquer pessoa pode, por exemplo, estudar o peso ao nascer em função das demais informações.

**Exercício** Você concorda com a publicação de dados? Porque? Não responda apenas com argumentação filosófica, cite pelo menos um exemplo real.

## 2.2 Algumas Fontes de Dados

O IBGE realiza várias pesquisas periódicas e algumas não periódicas. Dentre as periódicas, temos a Pesquisa Nacional por Amostragem de Domicílios - PNAD, com periodicidade anual. Esta pesquisa obtém basicamente dados demográficos e sócio-econômicos. Em alguns anos foram incluídas questões de saúde, de segurança, entre outras. A política de disponibilização dos dados de PNADS tem sido a de disponibilizar microdados, ocultando algumas variáveis que permitam a identificação de pessoas, e disponibilizar estimativas.

O censo populacional feito pelo IBGE tem periodicidade decenal, apenas em 1990 não foi realizado mas foi realizado um censo populacional em 1991. Neste caso, são disponibilizados muitos dados com referência geográfica, em nível estadual, municipal, bairros (para alguns municípios) ou setor sensitário. Ou seja, o nível territorial é bastante detalhado. Além do censo populacional, há outros, tais como o censo agropecuário anual, do qual são disponibilizados dados a nível de município.

O Ministério da Saúde (MS) também tem disponibilizado sistematicamente uma variedade enorme de informações de saúde com referência geográfica. São disponibilizados microdados do registro de nascimento e morte, microdados de atendimento ambulatorial, internações, etc. Nos dados de nascimento e morte há referência de município de residência e de ocorrência. Nos dados de atendimento ambulatorial e de internações há também o CEP da residência. O MS disponibiliza seus dados através de seu *site* <http://www.datasus.gov.br> (DATASUS).

No DATAUS também são disponibilizados dados tabulados, tais como número de nascidos vivos por município em cada ano. Ou seja, para visualizar um mapa de um estado dividido por municípios, colorindo cada município em função da mortalidade infantil, podemos vamos usar os dados do DATASUS tabulados por município.

A maioria dos dados são disponibilizados após algum período. Por exemplo, os dados de PNADs do IBGE e de nascimento e morte do DATASUS demoram aproximadamente dois anos para serem disponibilizados. Há dados que demoram menos tempo, tais como os dados de internações, que demoram uns dois meses para serem disponibilizados.

O Instituto Nacional de Pesquisas Espaciais (INPE) também tem uma política de disponibilizar dados com referência espacial. É possível, por exemplo, obter dados

diários de estações meteorológicas automáticas. Esses dados demoram poucos dias para serem disponibilizados.

A NASA disponibiliza dados obtidos de imagens de satélites e de radares embarcados em satélites. Esses dados são globais e podem ser obtidos para uma região do globo. Podemos por exemplo, obter a estimativa da precipitação acumulada a cada 90 minutos em cada quadrado de aproximadamente 2km de lado. Esses dados demoram poucas horas para serem disponibilizados.

**Exercício** Você sabe de alguma outra fonte de dados? Qual? Como se obtém esses dados?

## 2.3 Obtenção dos Dados

A maioria dos dados são obtidos a partir da página de internet do órgão que disponibiliza os dados. Geralmente há uma página específica para *download* dos dados. Nessa página geralmente há alguma forma de consulta de forma a se escolher a variável, período de tempo, abrangência ou nível geográfico, etc. A maioria deles está num formato simples de serem importados em *softwares* de análise estatística, tais como o formato texto.

### 2.3.1 Dados do IBGE

Na página principal do IBGE podemos escolher obter dados tabulados a partir do SIDRA (Sistema IBGE de Recuperação Automática). Basta clicar em <Banco de Dados> e depois em <SIDRA>. Nosso objetivo é fazer visualização espacial dos dados, neste caso vamos obter dados do Censo, cujos dados tabulados estão disponíveis até ao nível de setor censitário. Assim, vamos escolher a seção <Censo Demográfico e Contagem> e escolher a tabela 1552, por exemplo, que é uma tabela de "População residente, por situação do domicílio e sexo, segundo a forma de declaração da idade e a idade".

A tabela 1552 é multidimensional (2x3x3x3x132x2x31784) onde o último nível possui 31784 dimensões, que são localizações geográficas. Vamos selecionar: em Variável: "População Residente"; em Situação de Domicílio: "Total"; em Sexo: "Total", em Forma de declaração de idade: "Total"; em Idade: "Total"; em Ano: "2000" e "2010"; e em Unidade Territorial: "Município". Vamos montar a tabela de forma que Ano fique na coluna e Município na linha.

Nesta última dimensão, vamos considerar a opção de "Município 2009 em ordem de UF e código (5565)". Além disto, vamos selecionar apenas os municípios de Minas Gerais. Para isto, vamos clicar em seleção avançada, escolher Grupar, escolher unidade da federação e escolher Minas Gerais.

Um detalhe importante é marcar, no cabeçalho da dimensão Unidade Territorial a opção <Exibir código>. Isto para que a coluna de código de município seja incluída. Esta coluna é necessária para que possamos, mais adiante, associar esses dados a um mapa e fazer a visualização espacial.

Após isso, vamos escolher a opção <Gravar> e escolher um nome para o arquivo. O arquivo salvo tem a extensão .csv. Agora, basta ler esse arquivo em **R** e fazer análises. Nós escolhemos o nome “pop20002010munMG” e o arquivo salvo foi “pop20002010munMG.csv”. O exemplo considerado pode ser importado diretamente (até o dia 21 de janeiro de 2012) para o **R** com

```
pop <- read.csv2("http://www.sidra.ibge.gov.br/
download/pop20002010munMG.csv", skip=7, nrow=854)
```

O IBGE também disponibiliza algumas variáveis tabuladas por setores censitários. No link [ftp://ftp.ibge.gov.br/Censos/Censo\\_Demografico\\_2000/Dados\\_do\\_Universo/Agregado\\_por\\_Setores\\_Censitarios/](ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2000/Dados_do_Universo/Agregado_por_Setores_Censitarios/) Temos um arquivo para cada UF dos dados do censo 2000.

Segundo o site do IBGE “Os agregados por setores censitários foram gerados a partir dos microdados do universo do Censo Demográfico 2000 e são formados por 21 planilhas de dados para cada Unidade da Federação, contendo mais de 3.200 variáveis.

Setor Censitário é unidade territorial de coleta das operações censitárias, definido pelo IBGE, com limites físicos identificados, em áreas contínuas e respeitando a divisão político-administrativa do Brasil.

O Território Nacional foi dividido em 215811 setores para a realização do Censo Demográfico de 2000.”

### 2.3.2 Dados do Datasus

A partir da página principal do Datasus, podemos obter facilmente dados tabulados ao nível de município. Também podemos acessar microdados de Nascimentos, de Mortalidade e microdados de alguns serviços prestados pelo Sistema Único de Saúde, tais como atendimento ambulatorial e internações. Os dados tabulados podem ser obtidos no formato de arquivos texto separados por “;”. Os microdados são disponibilizados num formato específico do DATASUS.

Por exemplo, para conseguir dados de nascidos vivos por municípios, podemos clicar em <Informações de Saúde>, depois em <Estatísticas Vitais>, depois selecionar, por exemplo, <Nascidos vivos - 1994 a 2009>. Após isso, ou selecionamos um estado no mapa que aparece ou selecionamos na caixa de seleção logo acima do mapa.

A partir da página aberta, podemos construir uma tabela de dados, selecionando a informação na linha, a informação na coluna e qual a variável, neste caso Nascimentos por residência da mãe ou Nascimentos por local de ocorrência. Como exemplo, vamos colocar Município na linha, Ano de nascimento na Coluna e vamos considerar Nascimentos por residência da mãe. Podemos selecionar mais de um ano, por exemplo, selecionar os anos 2008 e 2009.

A seguir, no final da página, escolhemos um modo de exibição. Podemos escolher <Tabelas com bordas>, e será exibida uma tabela com a identificação do município (código e nome) e o número de nascidos vivos em cada um dos anos selecionados.



Ao final dessa tabela, temos a opção de “Copia com .CSV”. Neste caso será salvo um arquivo do tipo texto separados por “;”.

Agora, vamos, a partir da página principal do datasus, selecionar um microdado de nascimento. Procure o termo 'sinasc' no buscador disponível na página principal. Assim, será localizada a página do Sistema de Informações de Nascidos Vivos. <http://www.datasus.gov.br/catalogo/sinasc.htm>. Nesta página, clique em <Arquivos>, depois em e <Arquivos de 1996 em diante> e depois em <Arquivos de declarações de nascidos vivos>. Neste ponto, é mostrada uma tabela com cada linha sendo uma UF e cada coluna sendo um ano.

No interior dessa tabela estão os links para os arquivos de declarações de nascimento. Os números representam o tamanho do arquivo em Kb. Posicionando o cursor na linha Minas Gerais e na coluna 2009, observamos que o link é <ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/NOV/dnres/dnmg2009.dbc>. O diretório <ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/NOV/dnres/> lista todos os arquivos disponíveis. É importante conhecer essa estrutura para facilitar download automático desses dados.

**Exercício** Entre num site que disponibiliza dados e encontre um dado (ou microdado). Você gostou?

## 2.4 Importação em R

Vamos considerar os formatos de texto separados por vírgulas (ou ponto e vírgula) e o formato DBC.

### 2.4.1 Arquivos CSV

O IBGE, por exemplo, disponibiliza dados em formato de texto separado por vírgulas - CSV (*comma separated ...*). Vale aqui uma nota sobre o formato CSV. O separador decimal inglês é “.” (ponto) enquanto que o nosso é “,” (vírgula). Por esse motivo há dois formatos CSV, um onde os campos são separados por “,” (vírgula), quando o separador decimal é ponto; e outro onde os campos são separados por “;” (ponto-e-vírgula), quando o separador decimal é “,” (vírgula).

Temos em **R** algumas funções que trabalham com arquivos do tipo texto delimitados: `read.table()`, `read.csv()`, `read.csv2()`, `read.delim()` e `read.delim2()`. Observando o help dessas funções, notamos vários argumentos. Estes argumentos dependem da estrutura do arquivo texto que desejamos importar.

O exemplo dos dados de população do IBGE, considerado na seção anterior, pode ser importado usando a função `read.csv2()`, pois trata-se de um texto com “;” como sendo o separador de campos e “,” como sendo o separador decimal. Esse arquivo pode ser lido diretamente (até o dia 21 de janeiro de 2012) para o **R** com

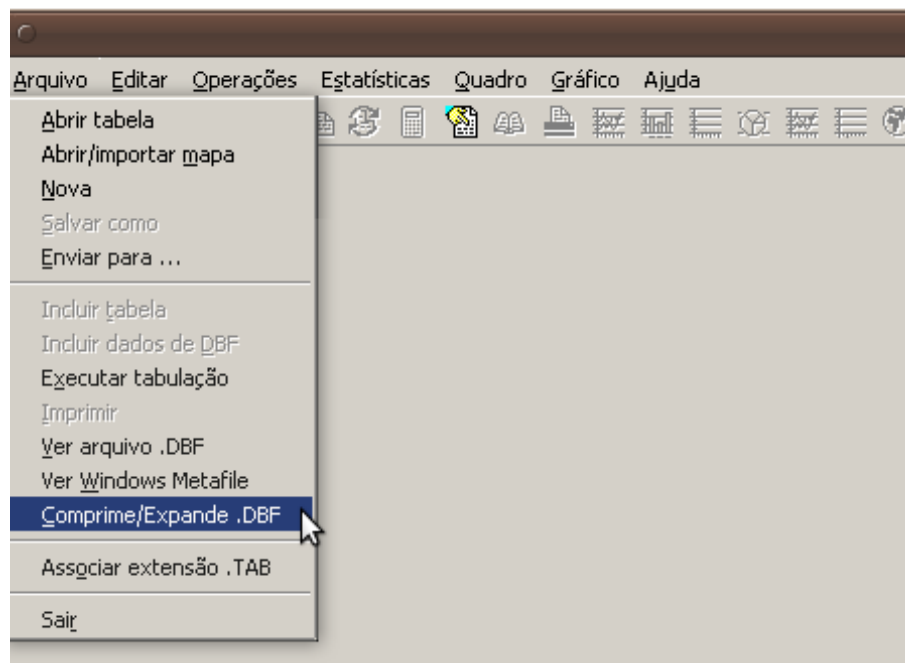
```
pop <- read.csv2("http://www.sidra.ibge.gov.br/
download/pop20002010munMG.csv", skip=7, nrows=854)
```

Onde, `skip=7` indica que vamos iniciar a leitura a partir da linha 8 do arquivo. Isto porque as primeiras 7 linhas contém a descrição/especificação dos dados. Usamos `nrows=854` porque nas últimas linhas contém várias linhas com notas explicativas sobre os dados, que não queremos importar.

### 2.4.2 Arquivos DBC

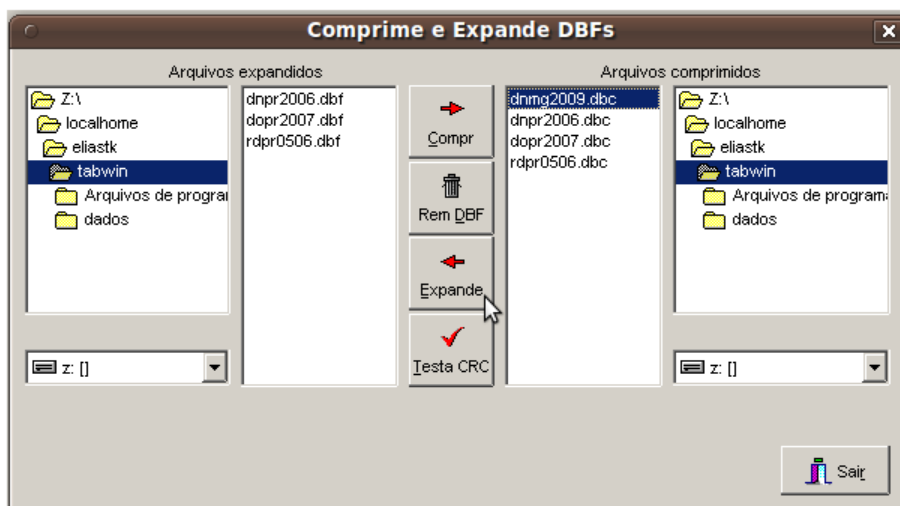
O DATASUS disponibiliza microdados em formato específico de arquivo, o formato DBC. Um arquivo DBC é um arquivo DBF comprimido. O DATASUS fornece um *software*, o TabWin, [?], que permite a expansão de arquivos DBC para o formato DBF e algumas análises exploratórias, incluindo tabulação e visualização espacial. As tabulações feitas pelo TabWin são implementadas em **R**. Portanto, para trabalhar com microdados do DATASUS é necessário usar o TabWin para expandir os arquivos DBC.

Após iniciar o TabWin, vamos no menu <Arquivo> e selecionamos a opção <Comprime/Expande .DBF>, como mostrado na figura a seguir



Na tela aberta, é necessário localizar o diretório onde está(ão), ou onde será(ão) salvo(s), o(os) arquivo(s) em formato .dbc. Também é necessário localizar o diretório onde estão, ou onde será(ão) salvo(s), o(s) arquivos .dbf.

Localizando os diretórios, basta selecionar o(s) arquivo(s) .dbc que serão expandidos para o formato .dbf e clicar em <Expandir>, conforme a figura a seguir



A importação em **R** de arquivos .dbf pode ser feita usando a função `read.dbf()` do pacote `foreign`, [?].

## 2.5 Exercícios

- Entre no site do IBGE e clique em "SIDRA". Na nova tela, clique em "Acervo". Selecione "Variáveis". Selecione "Salário". Selecione "Salários (Cadastro Central de Empresas)". Selecione "Tabela 1735 - Dados gerais (...) e Municípios". Obtenha o "Pessoal ocupado total", "Pessoal ocupado assalariado" e "Salários - decimais:3/0" para os anos de 1996 a 2006 para o município de Belo Horizonte.
  - Importe esses dados em **R**.
  - Faça um resumo descritivo e gráfico desses dados.
  - Calcule o salário por pessoa assalariada. Faça um resumo descritivo e gráfico desse dado.
- Entre no site do IBGE, clique em "SIDRA". Na nova tela, clique em "Acervo". Selecione "Níveis Territoriais". Selecione "Bairro" e em seguida "População". Selecione "Média de moradores por domicílio". Selecione "Tabela 156 - Média de moradores por domicílio". Obtenha o número médio de moradores por domicílios dos 18 bairros do município Araucária e para os 75 bairros do município de Curitiba
  - Importe esses dados em **R**.
  - Faça uma análise descritiva desses dados
  - Teste se o número médio de moradores por domicílios nos bairros de Araucária difere do número médio de moradores dos bairros de Curitiba
- Entre no site do DATASUS. Clique em "Informações de Saúde". Na lista de opções aberta, clique em "Estatísticas Vitais". Selecione "Nascidos vivos -

1994 a 2007”. No mapa mostrado ao lado, selecione o estado de Minas Gerais. Obtenha o “número de nascidos vivos por município de residência da mãe”por “Duração da Gestação”no ano de 2009 nos municípios de Curitiba e Ouro Preto.

- (a) Importe essa tabela em **R**.
  - (b) Existe associação entre “Duração da Gestação”e município? (Desconsidere os ignorados se houver.)
4. Entre no site do DATASUS. Clique em “Informações de Saúde”. Na lista de opções aberta, clique em “Estatísticas Vitais”. Selecione “Nascidos vivos - 1994 a 2007”. No mapa mostrado ao lado, selecione o estado de Minas Gerais. Obtenha o número de nascidos vivos por local de residência da mãe por “Peso ao Nascer”no ano de 2009 em todos os municípios de Minas Gerais.
- (a) Importe esses dados em **R**.
  - (b) Faça uma análise descritiva
  - (c) Identifique a categoria de “Peso ao Nascer”que possui o maior número de nascidos vivos no estado
  - (d) Calcule a proporção dos nascidos vivos nessa categoria para cada município e faça uma análise descritiva (Desconsidere os ignorados se houver).
5. Entre no site de sistema de informações de mortalidade <http://www.datasus.gov.br/catalogo/sim.htm>.

**NOTA** : Considere o arquivo de documentação disponível nessa página.

- (a) Faça o download e um arquivo de declarações de mortalidade de MG em 2009
- (b) Expanda o arquivo para .dbf usando o TabWin
- (c) Importe esses dados para o **R**
- (d) Quantos homens e quantas mulheres foram a óbito?
- (e) Calcule a idade média para homens e mulheres separadamente

# Capítulo 3

## Mapas

Um mapa de áreas (municípios, estados) é, basicamente, um conjunto de polígonos. Porém, geralmente temos atributos (variáveis) associados aos polígonos, tais como PIB, População Residente, IDH, etc. Um formato padrão de mapa em formato digital é o *shapefile*, que é um conjunto de pelo menos três arquivos: um arquivo com os polígonos (com extensão .shp), um arquivo com os atributos (com extensão .dbf), e um arquivo com índices (com extensão .shx). Outro arquivo que pode estar incluso num *shapefile* é o arquivo indicando o sistema de projeção das coordenadas dos polígonos, com extensão .prj.

Há outros formatos usados para mapas, que também podem importados em R. Um mais recente é o formato kml, usado pelo *google earth*.

### 3.1 Fonte de Mapas Territoriais no Brasil

O IBGE disponibiliza um conjunto de milhares de mapas. Há mapas do Brasil dividido por municípios e mapas de cada município dividido por setores censitários. Os polígonos de cada área (município/setor censitário), são representados por um conjunto de pontos. Quanto mais pontos utilizados na representação de um polígono, mais apropriado será o mapa para a produção de mapas em alta resolução gráfica. O IBGE disponibiliza mapas para três diferentes resoluções.

Um polígono pode estar representados em diferentes projeções ou cartográficas, [?]. O IBGE disponibiliza mapas em duas projeções: Projeção Geográfica e Projeção Policonica. Além disso, os mapas são disponibilizados em dois formatos de arquivo diferentes: *shapefiles* e *Mge\_Dgn*. O mapa de municípios está disponível em arquivo para o Brasil todo, por Região e por Unidade da Federação. Além disso, para cada município está disponível um *shapefiles* da divisão do município em setores censitários.

Vamos, por exemplo, obter o mapa de Minas Gerais subdividido em municípios. Na página principal do IBGE <http://www.ibge.gov.br>, clicamos em <Geociências>, depois em <Mapeamento das unidades territoriais>, depois em <Produtos> e depois em <Malha municipal digital 2007>. Na nova janela, escolhemos E2500 (entre as opções E500, E100 e E2500), que é a menor resolução, pois não estamos interessados num produto de maior resolução. Escolhemos a projeção geográfica

<Proj\_Geográfica>, escolhemos o formato *shapefiles* <ArcView\_shp>, escolhemos o nível territorial Unidade da Federação <UF> e escolhemos o estado de Minas Gerais <MG>.

Assim, chegamos no seguinte diretório `ftp://geoftp.ibge.gov.br/mapas/malhas_digitais/municipio_2005/E2500/Proj_Geografica/ArcView_Shp/Uf/MG/` E podemos fazer o download dos cinco arquivos que compoem o *shapefiles* com

```
d <- 'ftp://geoftp.ibge.gov.br/mapas/malhas_digitais/municipio_2005/E2500/Proj_G
for (e in c('dbf', 'sbn', 'sbn', 'shp', 'shx'))
  download.file(paste(d, '31mu2500gc.', e, sep=''),
                paste('../..maps/31mu2500gc.', e, sep=''))
```

## 3.2 Importando um *shapefiles* em R

Há mais de um pacote para leitura de mapas em R [?]. Nós vamos ler *shapefiles* usando o pacote `mapproj` [?]. Este pacote usa as classes de dados espaciais definidas no pacote `sp` [?].

Vamos considerar o mapa obtido na seção anterior. Inicialmente carregamos o pacote `mapproj`.

```
> require(mapproj)
```

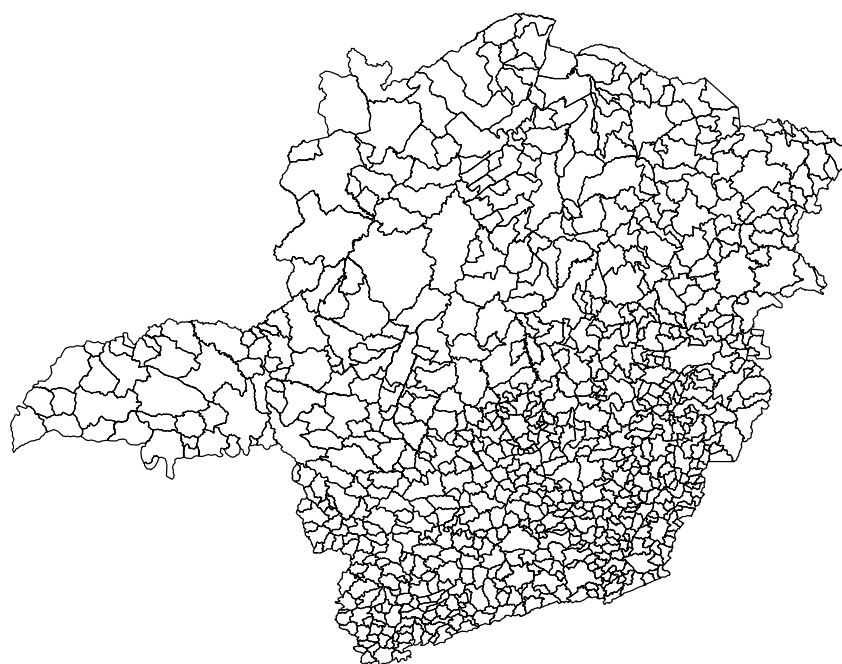
Vamos considerar que o conjunto de arquivos do *shapefiles* está do diretório “mapas” do diretório corrente. Vamos usar a função `readShapePoly()` para ler o mapa.

```
> mg <- readShapePoly("../..maps/31mu2500gc")
```

O objeto `mg` contém o conjunto de polígonos dos 853 municípios de Minas Gerais e uma tabela com alguns atributos desses municípios, isto é, um `data.frame` com 853 linhas. Podemos visualizar esse mapa simplesmente fazendo

```
> par(mar = c(0, 0, 0, 0))
> plot(mg)
```

onde `par(mar=c(0,0,0,0))` foi usado para tirar as margens da janela gráfica, aumentando sua área útil. O mapa produzido por este comando está na figura a seguir.



### 3.3 Georeferenciamento Espacialmente Discreto

Algumas vezes temos uma tabela com informações/atributos de áreas geográficas (municípios por exemplo) e gostaríamos de plotar essas informações num mapa. O georeferenciamento espacialmente discreto nada mais é que associar essa tabela de atributos ao mapa. Vamos considerar os dados de população do capítulo anterior e o mapa da seção anterior.

Na tabela de atributos do mapa temos uma coluna denominada GEOCODIGO, que é o código de cada município do mapa.

```
> names(mg)
[1] "GEOCODIGO" "NOME"      "UF"
[4] "ID_UF"     "REGIAO"   "MESOREGIAO"
[7] "MICROREGIA" "LATITUDE" "LONGITUDE"
[10] "SEDE"
```

Como podemos ver a seguir, esse código é formado por sete dígitos

Na tabela dos dados de população temos o código de município como parte da primeira coluna, isto é, os sete primeiros dígitos dessa coluna,

```
> head(pop)
      X      X2000      X2010
1      1 - Brasil 169799170 190755799
2 3100104 - Abadia dos Dourados - MG      6446      6704
3      3100203 - Abaeté - MG      22360     22690
4      3100302 - Abre Campo - MG     13348     13311
5      3100401 - Acaiaca - MG      3889      3920
6      3100500 - Açucena - MG     11489     10276
```

O que precisamos é, simplesmente, colocar os dados na mesma ordem do mapa. Isso será feito considerando o código do município. Inicialmente, vamos criar uma variável que é o código de município obtido da tabela dos dados.

```
> codm <- substr(as.character(pop[,1]), 1, 7)
```

Vamos criar uma nova tabela contendo o código, o nome, a mesoregião, a micro região, a população em 2000 e a população em 2010.

```
iodad <- merge(mg@data[,c(1, 2, 6:7)],
              data.frame(cod=codm, pop[, -1]),
              by.x='GEOCODIGO', by.y='cod', sort=FALSE)
```

e vamos convervir se esta tabela esta na mesma ordem do mapa.

```
> all.equal(mg@data[,1], iodad[,1])
[1] TRUE
```

Agora, podemos visualizar os dados no mapa, isto é, criar um mapa temático.

### 3.4 Mapa temático

Um mapa temático é um mapa de uma região geográfica colorida de acordo com o valor de um atributo. Por exemplo, podemos colorir o mapa de Minas Gerais dividido por município em tons de cinza, onde cinza mais escuro indica maior mortalidade infantil. Nesta seção vamos aprender a criar um mapa temático.

Vamos considerar os dados de população em 2000 e 2010 da seção anterior. Vamos considerar que queremos colorir o mapa de forma que os municípios que tiveram a sua população aumentada estejam de uma cor e os demais de outra cor.

Inicialmente, vamos criar uma variável que indica se a população de cada município aumentou ou não



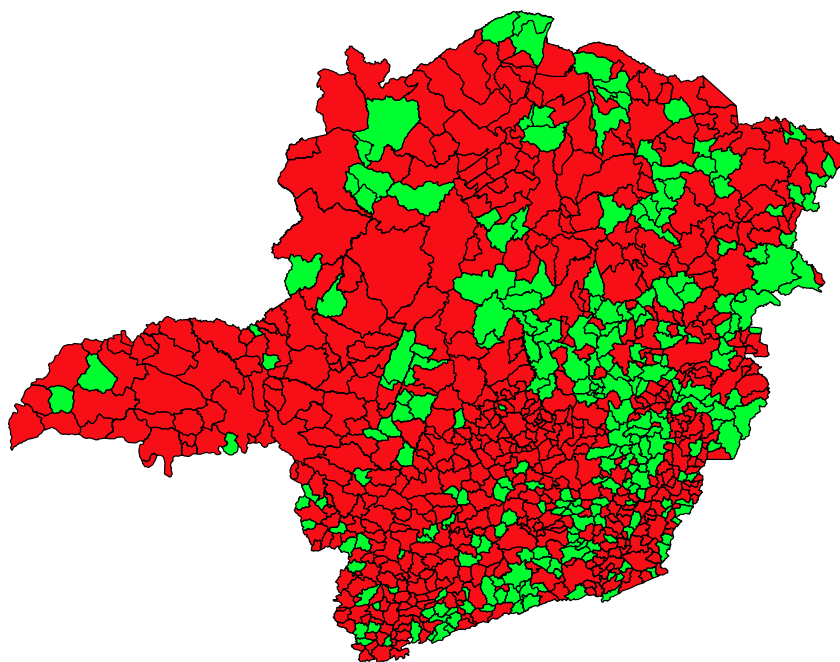
```
> table(aum <- odad[,6]>odad[,5])
```

```
FALSE TRUE  
 259   594
```

Podemos visualizar o mapa colorindo os municípios que tiveram aumento em vermelho e os demais em verde, assim

```
> plot(mg, col=ifelse(aum, 'red', 'green'))
```

resultando o mapa da figura a seguir



Podemos criar um mapa com mais cores, por exemplo, tons diferentes de acordo com o percentual de aumento da população. Vamos criar inicialmente o vetor que indica o aumento percentual da população

```
> summary(paum <- odad[,6]/odad[,5] - 1)
   Min.  1st Qu.  Median    Mean  3rd Qu.    Max.
-0.23800 -0.01068  0.04325  0.05540  0.10650  0.96810
```

Vamos categorizar essa variável em cinco níveis: menor que -0.02; de -0.02; de 0.02 a 0.05; de 0.05 a 0.15; e maiores que 0.15. Assim,

```
> br <- c(-Inf, -0.02, 0.02, 0.05, 0.15, Inf)
> table(c15 <- findInterval(paum, br))

 1   2   3   4   5
187 149 116 280 121
```

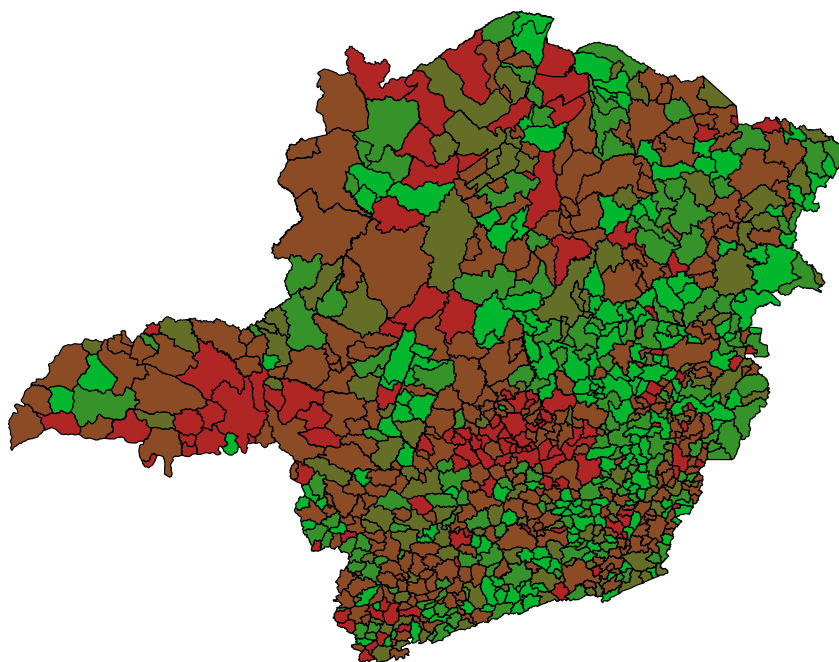
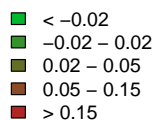
Agora vamos definir as cores. Vamos usar tons RGB (red, green e blue) de forma que as primeiras cores tenham um tom mais avermelhado e as últimas, um tom mais esverdeado.

```
> cores5 <- rgb(1:5/7, 5:1/7, 0.1)
```

O mapa com mais de duas cores pode ser feito assim

```
> plot(mg, col=cores5[c15])
> legend('topleft', leglabs(br, '<', '>'),
+       fill=cores5, bty='n')
```

Adicionamos uma legenda para as cores, resultando num mapa temático conforme a figura a seguir



## 3.5 Exercícios

1. Entre no site do IBGE. Clique em Geociências. Clique em Mapeamento das unidades territoriais. Clique em Produtos. Clique em Malha municipal digital 2007 (clique no disquete de 2007). Na nova janela escolha E500, E100 ou E2500. Escolha uma projeção. (Proj\_Geografica por exemplo). Escolha um formato. (Recomendo ArcView\_shp) Escolha o nível territorial de UF. Escolha a unidade territorial MG.

(a) Importe esse mapa em **R**.

- (b) Visualize esse mapa.
2. Considere o mapa de Minas Gerais dividido em Municípios.
    - (a) Encontre o município de Belo Horizonte e visualize-o.
    - (b) Encontre a microregião de Belo Horizonte e visualize-a.
    - (c) Adicione o nome a cada município.
  3. Considere o mapa de Minas Gerais dividido em Municípios.
    - (a) Encontre a lista de vizinhança.
    - (b) Visualize esse grafo.
    - (c) Encontre quem são os municípios vizinhos de Belo Horizonte.
  4. Considere os dados do exercício 3 do capítulo anterior
    - (a) Associe esses dados ao mapa de Minas Gerais
    - (b) Faça um mapa temático colorindo-o de acordo com a proporção calculada
  5. Considere os exercício anterior.
    - (a) Exporte o mapa para o formato *shapefile*, incluindo uma coluna com a proporção calculada.

# Capítulo 4

## Georeferenciamento

Eventualmente trabalhamos com dados que incluem endereço. Esse endereço pode ser o logradouro (Rua, Avenida, etc.) com o número ou apenas o CEP. Os dados de internações do DATASUS, por exemplo, contém o CEP de residência dos pacientes. Neste caso, um CEP pode referir-se a todas as residências de uma rua inteira, ou seja, a localização não é exata. Neste capítulo, vamos mostrar como fazer o georeferenciamento de endereços usando uma API de georeferenciamento do google-Maps. Também vamos obter uma imagem de mapa de ruas e sobrepor endereços georeferenciados a essa imagem.

### 4.1 Georeferenciamento de Logradouro

Um endereço especificado por logradouro é completo quando temos o tipo de logradouro (Av., Rua, etc.), o nome do logradouro, o número do endereço no logradouro, a cidade, estado e país. Suponha que temos alguns endereços, guardados na forma de logradouro, nome do logradouro, número e cidade. No ambiente R, teríamos, por exemplo:

```
> ltipo <- c("av", "rua", NA)
> lnome <- c("Amazonas", "curitiba", "prefeitura municipal")
> enum <- c(500, 1101, NA)
```

todos estes em Belo Horizonte. Note que o terceiro endereço é simplesmente "prefeitura", sem logradouro.

Nós vamos usar uma API do `google.maps` para fazer o georeferenciamento desses endereços, isto é, obter a latitude e longitude desses endereços. Para entender o que essa API faz, digite

```
http://maps.google.com/maps/geo?q=
rua+curitiba+100+Belo+Horizonte
&output=csv&sensor=true_or_false&key=abcdefg
```

num navegador de internet.

O resultado exibido deve ser

```
200,8,-19.9133353,-43.9398656
```

que nada mais é que uma linha com quatro colunas separadas por “,”. A primeira coluna é um código de sucesso da operação (200 é sucesso), cujo significado esta em

```
http://code.google.com/intl/pt-BR/apis/maps/documentation/
javascript/v2/reference.html#GGeoStatusCode
```

A segunda coluna é um código de qualidade (acurácia) do georeferenciamento cujo significado está em

```
http://code.google.com/intl/pt-BR/apis/maps/documentation/
javascript/v2/reference.html#GGeoAddressAccuracy
```

As duas últimas colunas são a latitude e longitude, respectivamente. O código de acurácia de georeferenciamento considera a divisão política dos EUA, isto é, estado, condado, cidade. Mas no Brasil, temos estado, município, bairro. Além disso, como veremos num exemplo, avenidas longas possuem mais de um CEP, sendo mais preciso neste caso georeferenciar por CEP que por logradouro.

Note que esse resultado é nada mais que um arquivo em formato de texto (muito simples) num servidor. Nós podemos ler esse arquivo no R com funções do tipo `read.table()`, `read.csv()`, etc. Particularmente eu uso a função `readLines()`, assim:

```
> url <- paste("http://maps.google.com/maps/geo?q=",
+             "rua+curitiba+100+Belo+Horizonte",
+             "&output=csv&sensor=true_or_false&key=abcdefg", sep="")
> readLines(url, warn=FALSE)

[1] "200,8,-19.9133353,-43.9398656"
```

que retorna uma *string*. Basta quebrar essa *string* nas “,”

```
> strsplit(readLines(url, warn=FALSE), ",")

[[1]]
[1] "200"      "8"        "-19.9133353"
[4] "-43.9398656"
```

e converter este resultado para número. Então, temos

```
> as.numeric(strsplit(readLines(url, warn=FALSE), ",")[[1]])

[1] 200.00000  8.00000 -19.91334 -43.93987
```

que pode ser armazenado num vetor.

Muitas vezes temos mais de um endereço para georeferenciar. Por isso, vamos definir uma função para fazer o georeferenciamento, a partir de um logradouro.

```

> fGetLatLonLog <- function(tipo, nome, num, mun, uf, pais) {
+   end <- paste(tipo, nome, num, mun, uf, pais, sep="+")
+   end <- gsub(" ", "+", end, fixed=TRUE)
+   end <- gsub("NA", "", end, fixed=TRUE)
+   end <- gsub("++", "+", end, fixed=TRUE)
+   end <- paste("http://maps.google.com/maps/geo?q=",
+               end,
+               "&output=csv&sensor=true_or_false&key=abcdefg", sep="")
+   end <- sapply(end, readLines, warn=FALSE)
+   end <- t(sapply(strsplit(end, ","), as.numeric))
+   colnames(end) <- c("Status", "Acuracia", "Latitude", "Longitude")
+   rownames(end) <- 1:nrow(end)
+   return(as.data.frame(end))
+ }

```

Aplicando essa função aos três endereços

```

> ll <- fGetLatLonLog(ltipo, lname, enum, "Belo Horizonte", "MG", "BR")
> ll

```

	Status	Acuracia	Latitude	Longitude
1	200	8	-19.91956	-43.93967
2	200	8	-19.92251	-43.94232
3	200	9	-19.92352	-43.93600

## 4.2 Localização de Endereço num Setor Censitário

Após georeferenciar, suponha que precisamos encontrar a qual setor censitário cada endereço pertence. Neste exemplo, precisamos do mapa de Belo Horizonte subdividido por setor censitário.

### 4.2.1 Mapa de Setores Censitários

O mapa de setores censitários de Belo Horizonte está disponível em

```

ftp://geoftp.ibge.gov.br/mapas/malhas_digitais/
setor_urbano_2000/MG/3106200/

```

O download do shapefile pode ser feito usando a função `download.file()` assim:

```

fd <- paste("ftp://geoftp.ibge.gov.br/mapas/",
"malhas_digitais/setor_urbano_2000/MG/3106200/", sep="")
for (e in c("DBF", "SHP", "PRJ", "SHX"))
  download.file(paste(fd, "3106200.", e, sep=""),
               paste("../maps/3106200.", tolower(e), sep=""))

```

Um shapefile pode ser lido em R com a função `readShapePoly()` do pacote `maptools`

```
> require(maptools)
> bh <- readShapePoly(".././maps/3106200")
```

e visualizado com `plot(bh)`. Vejamos as coordenadas dos centroides dos primeiros três setores

```
> head(coordinates(bh), 3)
      [,1] [,2]
[1,] 608890.0 7797736
[2,] 602636.0 7797562
[3,] 605011.1 7794975
```

Note que as coordenadas desse mapa estão projetadas, isto é, os valores não estão em graus de latitude e longitude.

### 4.2.2 Projeção Cartográfica

Esse mapa está numa projeção de coordenadas de forma que a distância entre pontos é dada em metros. Assim, ao fazer

```
> dist(head(coordinates(bh), 3))
      1      2
2 6256.470
3 4761.347 3512.260
```

obtemos a distância em metros entre os três primeiros setores censitários.

Ao georeferenciar os endereços nós obtemos as coordenadas em ângulos de latitude e longitude. Precisamos converter essas coordenadas para a mesma projeção do mapa considerado, para que possamos trabalhar com os endereços e esse mapa ao mesmo tempo.

A partir do arquivo com extensão `.prj`, obtemos qual é a projeção desse mapa. A string em formato PROJ4 para a projeção do mapa considerado é

```
+proj=utm +zone=23 +south +ellps=GRS67 +units=m +no_defs
```

Então, vamos usar a função `project()` do pacote `proj4` para fazer essa conversão de coordenadas. Precisamos informar uma matrix com duas colunas e a string no padrão PROJ4 para essa função. As duas colunas devem ser na ordem Longitude (eixo x) e Latitude (eixo y).

```
> require(rgdal)
> stringproj <- "+proj=utm +zone=23 +south +ellps=intl +units=m +no_defs"
> ##"+proj=utm +zone=23 +south +ellps=aust_SA +units=m +no_defs"
> ##"+proj=utm +zone=23 +south +ellps=intl +units=m +no_defs"
> llxy <- project(as.matrix(ll[, 4:3]), stringproj)
> llxy
```



```

      [,1] [,2]
[1,] 610981.8 7797043
[2,] 610702.6 7796718
[3,] 611362.8 7796602

```

Agora, podemos calcular distâncias em metros entre esses pontos georeferenciados,

```

> dist(data.frame(llxy))

      1      2
2 428.2997
3 582.8530 670.3408

```

### 4.2.3 Endereço Pontual em Setor Censitários

Podemos verificar a qual setor censitário pertencem esses três pontos. Inicialmente, vamos converter o objeto `llxy` num formato adequado

```

> llxy <- data.frame(x=llxy[,1], y=llxy[,2])
> coordinates(llxy) = ~x+y
> class(llxy)

```

```

[1] "SpatialPoints"
attr(,"package")
[1] "sp"

```

Após isso, usamos a função `over()` do pacote `sp` para encontrar os polígonos do mapa que contém esses pontos.

```

> o <- over(llxy, bh)
> o

      ID_ NAME1_ NAME2_ PARTS_ POINTS_
1 310620005620183 <NA> <NA>      1      13
2 310620005620218 <NA> <NA>      1       9
3 310620005620180 <NA> <NA>      1      15
      LENGTH_ AREA_
1 1.0348060 0.06056688
2 0.6474293 0.01763850
3 0.9723974 0.02667310

```

Precisamos saber a posição desses polígonos no mapa

```

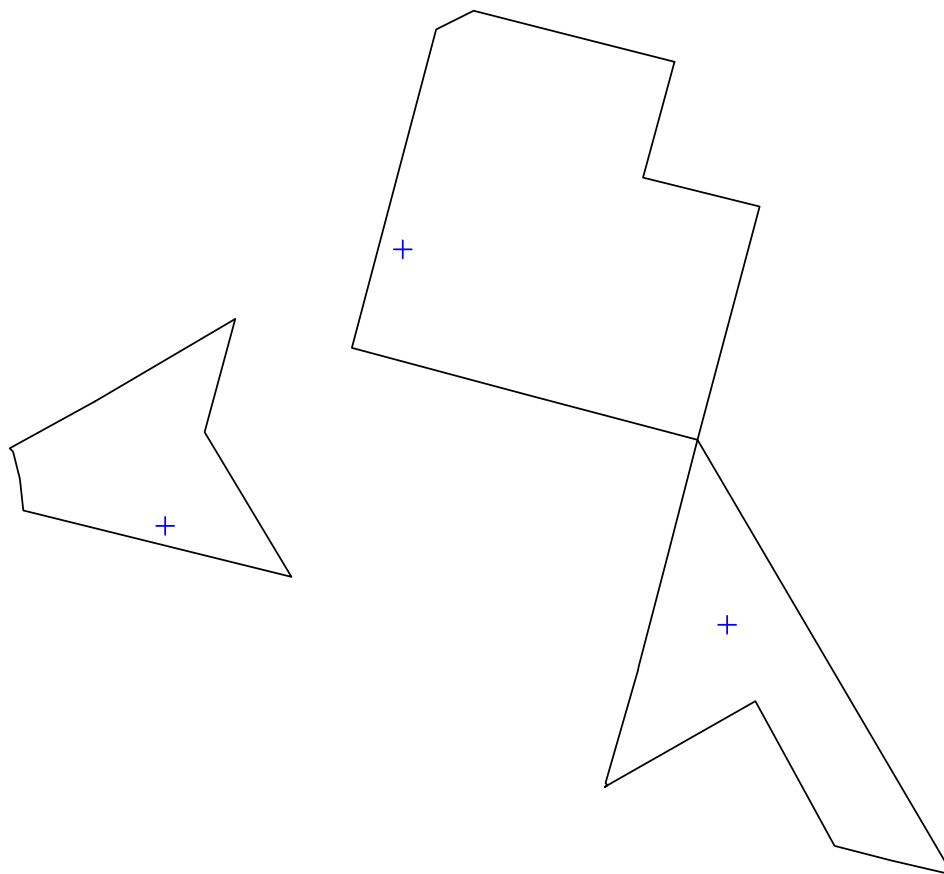
> id <- pmatch(o$ID_, bh$ID_, duplicates.ok=TRUE)
> id

[1] 1573 1562 1576

```

e após isso, usar essa informação. Por exemplo, podemos encontrar o sub-mapa desses setores e visualizar

```
> sbh <- bh[id,]  
> par(mar=c(0,0,0,0))  
> plot(sbh)  
> points(llxy, pch=3, col=4)
```



### 4.3 Georeferenciamento de CEP

Um interesse particular é fazer o georeferenciamento de CEPs. Há um Web Service que fornece o logradouro a partir de CEP, no site <http://cep.republicavirtual.com.br/>. Para entender o resultado digite, por exemplo,

```
http://cep.republicavirtual.com.br/web_cep.php?  
cep=30170120&formato=query_string
```

num navegador de internet. Há outras opções de formatos, tais como o xml.

Esse resultado nada mais é que um arquivo texto com uma linha. Podemos trazê-lo para o ambiente R usando uma função de leitura de arquivos em formato texto. Podemos criar uma função em R que extraia o logradouro a partir desse resultado. Com o logradouro, já sabemos como usar a API do googleMaps para fazer o georeferenciamento.

Existe uma limitação em usar o Web Service do republicavirtual. Esse serviço limita-se a um número de consultas por IP. Existe a possibilidade de fazer o download da base de CEPs disponíveis nesse site e trabalhar com ela localmente.

Outra opção é georeferenciar um CEP diretamente usando a API de georeferenciamento do googleMaps. Para isso, basta alterarmos a forma de consulta que definimos anteriormente. Podemos fazer assim

```
> fGetLatLonCEP <- function(cep, mun, uf, pais) {
+   end <- paste(cep, mun, uf, pais, sep="+")
+   end <- gsub(" ", "+", end, fixed=TRUE)
+   end <- gsub("NA", "", end, fixed=TRUE)
+   end <- gsub("++", "+", end, fixed=TRUE)
+   end<- paste("http://maps.google.com/maps/geo?q=cep=",
+             end,
+             "&output=csv&sensor=true_or_false&key=abcdefg", sep="")
+   end <- sapply(end, readLines, warn=FALSE)
+   end <- t(sapply(strsplit(end, ","), as.numeric))
+   colnames(end) <- c("Status", "Acuracia", "Latitude", "Longitude")
+   rownames(end) <- 1:nrow(end)
+   return(as.data.frame(end))
+ }
> ceps3 <- c(30180001, 30170120, 30130003)
> l12 <- fGetLatLonCEP(ceps3, "Belo Horizonte", "MG", "Brasil")
> l12

  Status Acuracia  Latitude Longitude
1     200         5 -19.91951 -43.94070
2     200         5 -19.92407 -43.94249
3     200         5 -19.92878 -43.92341

> l12xy <- project(as.matrix(l12[, 4:3]), stringproj)
> l12xy

      [,1] [,2]
[1,] 610874.2 7797048
[2,] 610683.8 7796546
[3,] 612677.4 7796011
```

Há uma grande vantagem em georeferenciar por CEP em vez de logradouro. Isso porque algumas ruas ou avenidas muito longas possuem mais de um CEP. No exemplo da Av. Amazonas, que é uma avenida muito longa, o georeferenciamento ao nível de rua é menos preciso que ao nível de CEP. Nesses casos, a coordenada geográfica é o ponto central do logradouro ou do CEP.

## 4.4 Imagens do GoogleMaps

O googleMaps é muito utilizado para a localização de um endereço. Esse endereço é mostrado num mapa com imagens de ruas ou numa imagem de satélite ou ambas sobrepostas. Uma imagem do googleMaps pode ser armazenada em formato de arquivos de imagens. Estes arquivos podem ser carregados em R onde podemos georeferenciar uma imagem. Assim, podemos trabalhar com mapas de setores censitários, endereços georeferenciados e imagens georeferenciadas.

### 4.4.1 Obtendo Imagens do GoogleMaps

Nesta seção nós vamos obter uma imagem do googleMaps e vizualizá-la usando o R. Para obter uma imagem, vamos utilizar o pacote GoogleMaps. Vamos obter uma imagem com centro o primeiro endereço georeferenciado

```
> require(RgoogleMaps)
> cooimg <- GetMap(center=unlist(l1[2,3:4]), zoom=16,
+   matype="roadmap", destfile="bhimg.png", format="png32")
```

A imagem obtida foi gravada num arquivo de imagem. Vamos agora usar um pacote para ler a imagem

```
> require(rgdal)
> imgbh <- readGDAL("bhimg.png", half.cell=c(0,0))
```

e vamos vizualizá-la com

```
> par(mar=c(0,0,0,0))
> image(imgbh, col=bpy.colors(50))
```



#### 4.4.2 Georeferenciamento de imagens

Nesta seção, vamos georeferenciar a imagem obtida e sobrepor os endereços georeferenciados anteriormente a essa imagem. Inicialmente vamos observar as coordenadas limites da imagem, armazenadas ao buscar a imagem no googleMaps

```
> cooimg[['BBOX']]
```

```
$l1
```

```
      lat      lon
[1,] -19.92898 -43.94917
```

```
$ur
```

```
      lat      lon
[1,] -19.91606 -43.93544
```

e vamos observar as coordenadas da imagem lida

```
> bbox(imgbh)
```

```

min    max
x -0.5 639.5
y -0.5 639.5

```

Vamos alterar os limites das coordenadas da imagem usando os limites da imagem buscada

```

> coobb <- t(do.call('rbind', cooimg[['BBOX']]))[2:1,]
> imgbh@bbox <- coobb

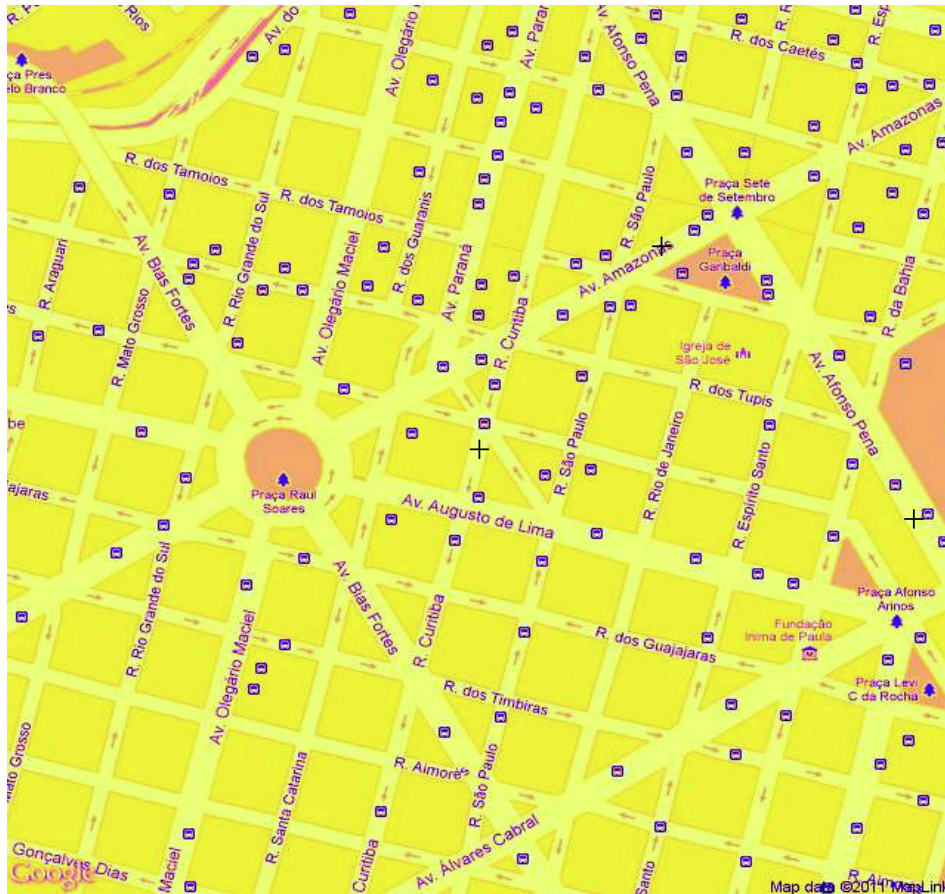
```

Assim, já podemos visualizar a imagem e sobrepor os três endereços georeferenciados.

```

> par(mar=c(0,0,0,0))
> image(imgbh, col=bpy.colors(50))
> points(l1[,4:3], pch=3)

```



Mas, para considerar o mapa de setores, precisamos projetar essa imagem no sistema de coordenadas do mapa de setores censitários. Para isso, podemos alterar novamente os limites da imagem



