Ecologia de Paisagens com R

Darren Norris

2023-05-10

Contents

Part I

Apresentação

Bem-vindos

Este é um trabalho em andamento do 1ª edição: **"Ecologia de Paisagens com R"**.

Este é um material introdutório destinado principalmente a estudantes de graduação e cursos de pós-graduação em ecologia e áreas correlatas.

O objetivo é de apresentar os capacidades e opções para desenvolver e integrar pesquisas na Ecologia da Paisagem no ambiente estatística de R.

Esperamos que ele seja utilizado tanto por quem quer se aprofundar em análises comumente utilizadas em Ecologia da Paisagem, quanto por quem não tem nenhuma ou poucas habilidades quantitativas.

Agradecimentos

Este livro não é apenas o resulto dos autores. Mas é o resultado de muitas pessoas na comunidade R e Ecologia da Paisagem no Brasil. Muito obrigado!

Prefácio à primeira edição

Há quem diga que a velocidade com que a tecnologia e a ciência avançam tende a tornar livros e manuais sobre métodos rapidamente obsoletos. A evolução dos computadores pessoais e a ampliação do acesso a estes e à Internet têm transformado o jeito como aprendemos e ensinamos.

Portanto, um livro aberto e disponível livremente que as pessoas possam compartilhar e contribuir torna-se uma opção cada vez mais relevante.

O conteudo e organização dos capítulos esta separado em partes.

O primeiro grupo – que inclui dos capítulos x ao y, inclui os aspectos mais gerais da estrutura do livro, seus objetivos e sobre o funcionamento da linguagem R. No segundo grupo de capítulos (do x ao y), temos contato com análises específicas e atualmente usadas em Ecologia da Paisasgem, incluindo ....

Introdução

Ecologia da Paisagem é uma disciplina empolgante que permite transformar dados brutos em compreensão, insight e conhecimento.

O que você vai aprender

Ecologia da Paisagem é um campo vasto e não há como dominar tudo lendo um único livro. Este livro visa fornecer uma base sólida nas ferramentas mais importantes e conhecimento suficiente para encontrar os recursos para aprender mais quando necessário. Um modelo das etapas de um projeto típico de Ecologia da Paisagem se parece com. . . ..

Como este livro está organizado

A descrição anterior das ferramentas da Ecologia da Paisagem é organizada aproximadamente de acordo com a ordem em que você as usa em uma análise (embora, é claro, você as itere várias vezes). Em nossa experiência, no entanto, aprender a importar e organizar os dados primeiro não é o ideal, porque 80% do tempo é rotineiro e chato e, nos outros 20% do tempo, é estranho e frustrante. Esse é um péssimo lugar para

começar a aprender um novo assunto! Em vez disso, começaremos com a visualização e transformação dos dados que já foram importados e organizados. Dessa forma, quando você ingerir e organizar seus próprios dados, sua motivação permanecerá alta porque você sabe que a dor vale o esforço.

Dentro de cada capítulo, tentamos aderir a um padrão consistente: comece com alguns exemplos motivadores para que você possa ver o quadro geral e depois mergulhe nos detalhes. Cada seção do livro é combinada com exercícios para ajudá-lo a praticar o que aprendeu. Embora possa ser tentador pular os exercícios, não há melhor maneira de aprender do que praticar em problemas reais.

O que você não vai aprender

Prerequisites

Fizemos algumas suposições sobre o que você já sabe para aproveitar ao máximo este livro. Você deve ser geralmente alfabetizado numericamente, e com conhecimento previsa de ecologia, geoprocessamento e uso de sistemas de informação geografica.

Você precisa de quatro coisas para executar o código deste livro: R, RStudio, uma coleção de pacotes R chamada **tidyverse** e um punhado de outros pacotes. Os pacotes são as unidades fundamentais do código R reproduzível. Eles incluem funções reutilizáveis, documentação que descreve como usá-los e dados de amostra.

R

Para fazer o download do R, acesse CRAN, a **c**omprehensive **R** **a**rchive **n**network, https://cloud.r-project.org. Uma nova versão principal do R é lançada uma vez por ano e há 2 a 3 versões secundárias a cada ano. É uma boa ideia atualizar regularmente. A atualização pode ser um pouco complicada, especialmente para as versões principais que exigem a reinstalação de todos os seus pacotes, mas adiar só piora as coisas. Recomendamos R 4.2.0 ou posterior para este livro.

RStudio

RStudio é um ambiente de desenvolvimento integrado, ou IDE, para programação R, que você pode baixar em https://posit.co/download/rstudio-desktop/. O RStudio é atualizado algumas vezes por ano e avisa automaticamente quando uma nova versão é lançada, para que não haja necessidade de verificar novamente. É uma boa ideia atualizar regularmente para aproveitar os melhores e mais recentes recursos. Para este livro, certifique-se de ter pelo menos o RStudio 2022.02.0.

O universo arrumado - tidyverse

Você também precisará instalar alguns pacotes do R. Um **pacote** do R é uma coleção de funções, dados e documentação que estende os recursos do R base. O uso de pacotes é a chave para o uso bem-sucedido do R. A maioria dos pacotes que você aprenderá neste livro faz parte do chamado tidyverse. Todos os pacotes no tidyverse compartilham uma filosofia comum de programação de dados e R e são projetados para trabalhar juntos.

Você pode instalar o tidyverse completo com uma única linha de código:

```
install.packages("tidyverse")
```

No seu computador, digite essa linha de código no console e pressione enter para executá-lo. R irá baixar os pacotes do CRAN e instalá-los em seu computador.

Você não poderá usar as funções, objetos ou arquivos de ajuda em um pacote até carregá-lo com `library()`. Depois de instalar um pacote, você pode carregá-lo usando a função `library()`:

```
library(tidyverse)
```

Isso diz a você que o tidyverse carrega nove pacotes: dplyr, forcats, ggplot2, lubridate, purrr, readr, stringr, tibble, alignr. Eles são considerados o **núcleo** do tidyverse porque você os usará em quase todas as análises.

Os pacotes no tidyverse mudam com bastante frequência. Você pode ver se há atualizações disponíveis executando `tidyverse_update()`.

Outros pacotes

Existem muitos outros pacotes excelentes que não fazem parte do tidyverse porque resolvem problemas em um domínio diferente ou são projetados com um conjunto diferente de princípios subjacentes. Isso não os torna melhores ou piores, apenas diferentes. Em outras palavras, o complemento do tidyverse não é o universo bagunçado, mas muitos outros universos de pacotes inter-relacionados. Ao lidar com mais projetos de Ecologia da Paisagem com R, você aprenderá novos pacotes e novas formas de pensar sobre os dados.

Usaremos outras pacotes de fora do tidyverse neste livro. Por exemplo, usaremos os seguintes pacotes porque eles fornecem conjuntos de funcões e dados interessantes para trabalharmos no processo de aprendizado de R:

```
install.packages(c("sp", "sf","raster", "mapview","tmap",
                   "terra", "kableExtra", "landscapemetrics"))
```

Part II

Escala e métricas

1 Escala

1.1 Apresentação

Nesta capitulo vamos entender a importância de escala na ecologia da paisagem através cálculos com a proporção de floresta. Durante o capitulo você aprenderá a

1. Alterar escala (resolução e extensão espacial),

2. Calcular a área de uma classe de habitat,

3. Desenvolve uma comparação multiescala.

É muito importante ficar claro para você o que é escala (e o que não é!) e qual a importância desse conceito na elaboração do desenho amostral, na coleta de dados, nas análises e na tomada de decisão. Nesse tutorial usaremos conteudo baseado no Capítulo 2 do livro (Fletcher and Fortin 2018) e "Tutorial Escala" do Dr. Alexandre Martensen.

1.1.1 Escala: breve definação

Todos os processos e padrões ecológicos têm uma dimensão temporal e espacial. Assim sendo, o conceito de escala não somente representar essas dimensões, mas também, ajudar nos apresentá-los de uma forma que facilite o entendimento sobre os processos e padrões sendo estudados.

Na ecologia o termo escala refere-se à dimensão ou domínio espaço-temporal de um processo ou padrão. Na ecologia da paisagem, a escala é frequentemente descrita por sua componentes: resolução e extensão.

- **Resolução**: menor unidade espacial de medida para um padrão ou processo.
- **Extensão**: descreve o comprimento ou tamanho de área sob investigação.

Resolução e extensão tendem a covariar – estudos com maior extensão tendem a ter resolução maiores também. Parte dessa covariância é prática: é difícil trabalhar em grandes extensões com dados coletados em tamanhos de resolução finos. No entanto, parte dessa covariância também é conceitual: muitas vezes em grandes extensões, podemos esperar que processos operando em resolução muito finos forneçam somente

"ruído" e não dados/informações relevantes sobre os sistemas. Como os desafios computacionais diminuíram e a disponibilidade de dados de alta resolução aumentou, a covariância entre resolução e extensão nas investigações diminuiu.

1.2 Pacotes e dados

Agora vamos olhar um exemplo do mundo real. Uma pequena amostra do Rio Araguari, perto de Porto Grande. O ponto central da raster é de longitude: -51.406312 latitude: 0.726236. Para visualizar o ponto no Google Earth: https://earthengine.google.com/timelapse#v=0.72154,-51.41543,11.8,latLng&t=2.24&ps=25&bt=19840101&et=20201231&startDwell=0&endDwell=0 .

Em geral é necessário baixar alguns pacotes para que possamos fazer as nossas análises. Precisamos os seguintes pacotes, que deve esta instalado antes:

- tidyverse,
- sf,
- mapview,
- tmap.

Carregar pacotes:

```
library(tidyverse)
library(sf)
library(terra)
library(mapview)
library(tmap)
```

O arquivo raster tem uma pequena amostra com a classificação da terra feito pela MapBiomas, que produz mapeamento anual da cobertura e uso da terra no Brasil desde 1985.

Baixar arquivo com os dados (formato ".tif"), link: https://github.com/darrennorris/gisdata/blob/master/inst/raster/amostra_mapbiomas_2020.tif . Lembrando-se de salvar o arquivo ("amostra_mapbiomas_2020.tif") em um local conhecido no seu computador.

Carregar o arquivo trabalhamos com o pacote terra. O pacote tem varios funçoes para a análise e modelagem de dados geográficos. Nós podemos ler os dados de cobertura da terra no arquivo ".tif" com a função `rast`.

```
#
ramostra <- rast(file.choose())
```

Plotar para verificar.

```
plot(ramostra)
```

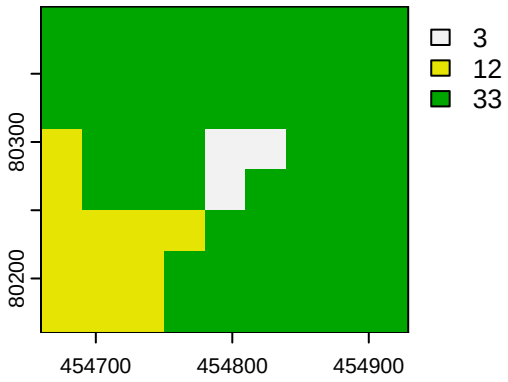

Figure 1: Mapbiomas 2020. Uma pequena amostra do Rio Araguari, perto de Porto Grande.

Podemos também verificar informações sobre o raster (metadados).

```
ramostra
```

```
## class       : SpatRaster
## dimensions  : 8, 9, 1  (nrow, ncol, nlyr)
## resolution  : 29.89281, 29.89281  (x, y)
## extent      : 454659.8, 454928.9, 80160.06, 80399.2  (xmin, xmax, ymin, ymax)
## coord. ref. : SIRGAS 2000 / UTM zone 22N (EPSG:31976)
## source      : amostra_mapbiomas_2020.tif
## name        : mapbiomas_2020
## min value   :              3
## max value   :             33
```

Isso nos mostra informações sobre escala espacial (resolução e extensão) e a sistema de coordenadas (SIRGAS 2000 / UTM zone 22N (EPSG:31976

$$https://epsg.io/31976$$

)). Além disso é possível obter informações específicas através de funções específicas.

```
# Obter informações sobre escala espacial
# resolução
res(ramostra)
# numero de colunas
ncol(ramostra)
# numero de linhas
nrow(ramostra)
```

1.2.0.1 Pergunta 1 **Com base nos resultados obtidos até agora em relação ao objeto raster ramostra, qual o tamanho do pixel em metros quadrados? Qual o tamnaho total da raster ramostra em hectares e quilometros quadrados?**

Vamos olhar o mapa de novo.

```
plot(ramostra)
```

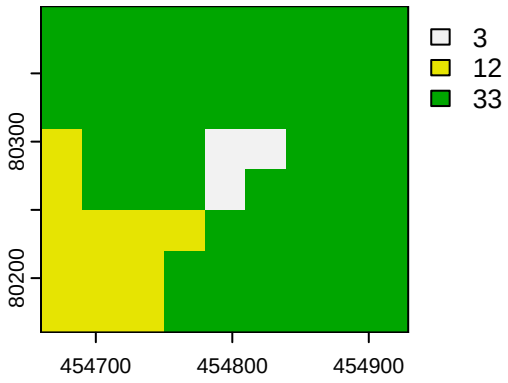

Figure 2: Mapbiomas 2020. Uma pequena amostra do Rio Araguari, perto de Porto Grande.

O mapa mostra três classes com valores de 3, 12 e 33. Lembrando, o objetivo principal não é de fazer mapas. Mas, a visualização dos dados é um passo importante para verificar e entender os padrões. Portanto, segue exemplo mostrando uma forma de visualizar o arquivo de raster como mapa.

Para entender o que os valores (3, 12, 33) representam no mundo real precisamos de uma referência (legenda). Para a MapBiomas Coleção 6, arquivo: Cod_Class_legenda_Col6_MapBiomas_BR.pdf. Existe também arquivos para fazer as mapas com cores corretas em QGIS ou ArcGIS.

Olhando a legenda (Cod_Class_legenda_Col6_MapBiomas_BR.pdf), sabemos que "3", "12" e "33" representem cobertura de "Formação Florestal", "Formação Campestre", e "Rio, Lago e Oceano". Então podemos fazer um mapa mostrando tais informações.

Daqui pra frente vamos aproveitar uma forma mais elegante de apresentar mapas e gráficos. Isso seria atraves a função "ggplot" (pacote ggplot2), que faz parte do "tidyverse". Mais exemplos no R cookbook : http://www.cookbook-r.com/Graphs/ .

E com mais exemplos de mapas e dados espaciais no R: sf e ggplot2 : https://www.r-spatial.org/r/2018/10/25/ggplot2-sf.html

Capitulo 9 no livro Geocomputation with R : https://geocompr.robinlovelace.net/adv-map.html

Primeiramente precisamos incluir as informações relevantes da legenda. Ou seja, incluir os nomes para cada valor de classe.

```
# legenda e cores na sequencia correta
classe_valor <- c(3, 12, 33)
classe_legenda <- c("Formação Florestal",
                    "Formação Campestre", "Rio, Lago e Oceano")
classe_cores <- c("#006400", "#B8AF4F", "#0000FF")
```

Agora podemos fazer o mapa com as classes e os cores seguindo o padrão recomendado pela MapBiomas para Coleção 6.

```
# Passo necessario para mostrar os valores
ramostra_df <- as.data.frame(ramostra, xy = TRUE)

ggplot(ramostra_df, aes(x=x, y=y)) +
```

```
geom_raster(aes(fill = factor(mapbiomas_2020))) +
scale_fill_manual("classe",
                    values = classe_cores,
                  labels = classe_legenda) +
coord_equal() +
geom_text(data = ramostra_df, aes(x = x, y = y,
              label = mapbiomas_2020)) +
theme(legend.position="top") +
guides(fill=guide_legend(nrow=2,byrow=TRUE))
```

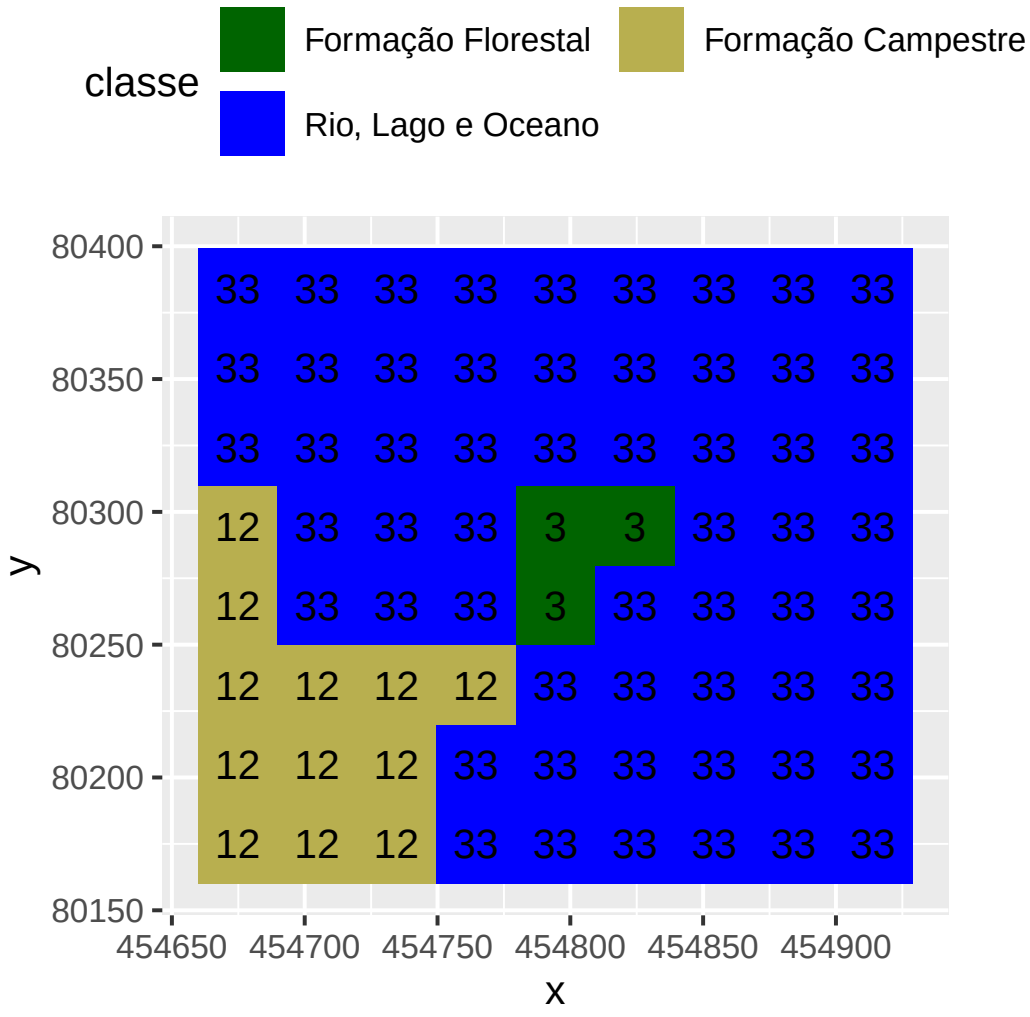

Figure 3: Paisagem com valores e classes de cobertura da terra. Mapbiomas 2020. Uma pequena amostra do Rio Araguari, perto de Porto Grande.

1.3 Alterando a resolução

Alterando a resolução serve como exemplo mostrando como os passos/etapas/cálculos mude dependendo o tipo de dados. Ou seja, é preciso adotar metodologias diferentes para dados categóricos (por exemplo classificação de cobertura da terra) e dados contínuos (por exemplo distância até rio).

Alterando a resolução às vezes seria necessário, por exemplo, quando preciso padronizar dados/imagens oriundos de fontes diferentes com resoluções diferentes e/ou para reduzir a complexidade da modelagem. Lembrando - em cada nível de resolução, são observáveis processos e padrões que não podem necessariamente ser inferidos daqueles abaixo ou acima.

Agora iremos degradar a resolução desses dados, ou seja, iremos alterar o tamanho dos pixels. Como exemplo, iremos juntar (agregar) 3 pixels em um único pixel. Como você acha que podemos fazer isso? Quais valores esse pixel que vai substituir os 3 originais deve ter? Existem diversas maneiras de se fazer isso, uma das formas é através da média.

```
ramostra_media<-aggregate(ramostra, fact=3, fun="mean")
ramostra_media <- resample(ramostra, ramostra_media)
```

Visualizar. Os valores calculados pela função não fazem sentido para uma classificação categórica.

```
# Tidy
as.data.frame(ramostra_media, xy = TRUE) %>%
  mutate(mapbiomas_2020 = round(mapbiomas_2020,1)) -> ramostra_media_df
# Plot
ggplot(ramostra_media_df, aes(x=x, y=y)) +
  geom_raster(aes(fill = factor(mapbiomas_2020))) +
  scale_fill_discrete("valor") +
  coord_equal() +
  geom_text(data = ramostra_media_df, aes(x = x, y = y,
                label = mapbiomas_2020))
```

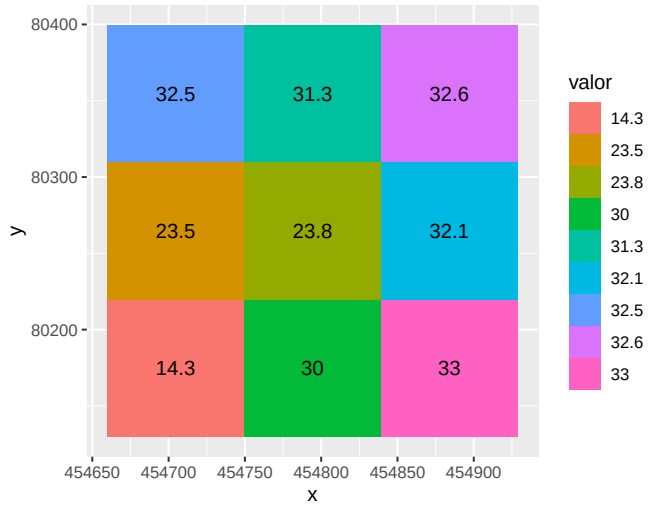

Figure 4: Agregação errado para dados categóricos. Uso da média cria valores categóricos errados e impossíveis.

Outra opção é utilizar o valor mais comum da área, o que é particularmente adequado quando temos um mapa categórico, como por exemplo floresta/não-floresta. Segue exemplo com o valor mais frequente (modal).

```
ramostra_modal<- aggregate(ramostra, fact=3, fun="modal")
ramostra_modal <- resample(ramostra, ramostra_modal, method="near")
```

Visualizar. Os valores calculados pela função são consistentes com o original e fazem sentido.

```
# Tidy
ramostra_modal_df <- as.data.frame(ramostra_modal, xy = TRUE)
# Plot
ggplot(ramostra_modal_df, aes(x=x, y=y)) +
  geom_raster(aes(fill = factor(mapbiomas_2020))) +
  scale_fill_manual("classe", values = classe_cores) +
  coord_equal() +
geom_text(data=ramostra_modal_df, aes(x=x, y=y, label=mapbiomas_2020))
```

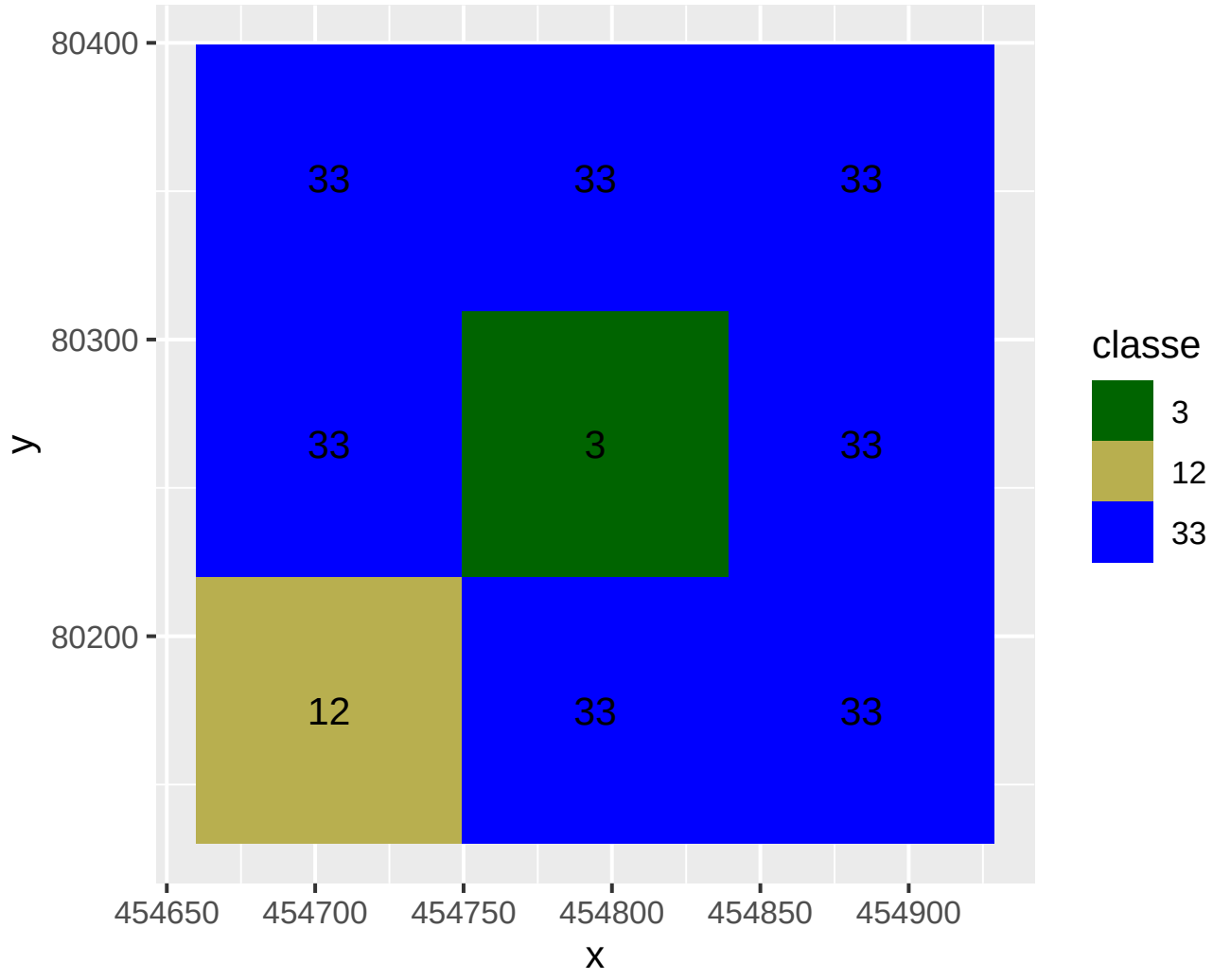

Figure 5: Agregação pela mais frequente.

Em cada nível de resolução, são observáveis processos e padrões que não podem necessariamente ser inferidos daqueles abaixo ou acima. Aqui por exemplo, mudamos a proporção de cobertura florestal em nossa pequeno paisagem quando juntamos 3 pixels em um único: a proporção de floresta moudou de 4% (3/72) para 11% (1/9). Ou seja, com cada passo mudamos a representção do mundo.

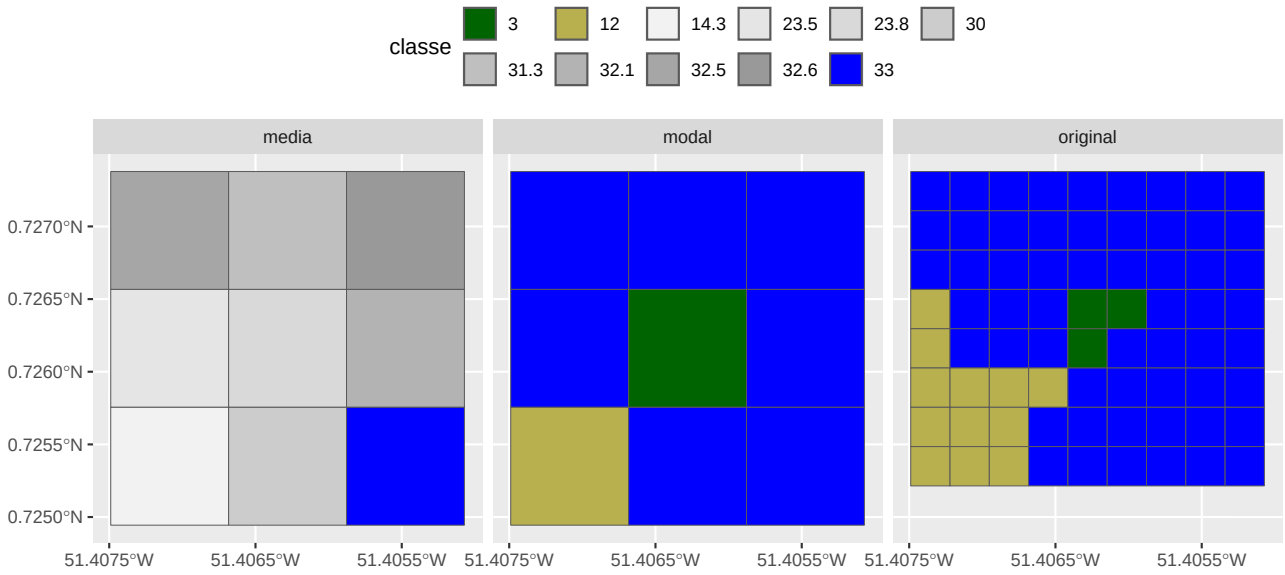

Figure 6: Mudanças causadas pela agregação.

1.3.0.1 Pergunta 2 **Confira o codigo e os resultados obtidos anteriormente, quando mudamos a resolução da ratser ramostra (por exemplo figuras 5.1, 5.2 e 5.3). Explique o que aconteceu. Como e porque moudou os valores em cada caso (média e modal)?**

1.4 Escala espacial e desenho amostral

Dado o papel que a escala pode desempenhar em nossa compreensão dos padrões e processos ecológicos, como escala deve ser considerada no desenho do estudo? Claramente, a resposta a esta pergunta irá variar dependendo dos fenômenos de interesse, mas ecologistas e estatísticos têm forneceu algumas orientações importantes. As questões-chave incluem o tamanho da unidade de amostragem (resoloção), o tipo de unidade de amostra e localizações da unidade de amostra, incluindo o espacamento entre as amostras (distância entre as amostras) e o tamanho da área de estudo.

Com a disponibilidade de imagens de satélite é possível responder questões importantes relacionadas ao desenho do estudo antes de qualquer trabalho de campo. Uma tecnica de geoprocessamento (bordas - Buffers) é um dos mais frequentemente adotados para quantificar escala espacial na ecologia da paisagem.

O objetivo é criar buffers circulares de diferentes extensões ao redor dos sitios de amostragem (pontos, pixels, manchas, transetos lineares etc). Aqui, vamos entender a escala em que a cobertura de floresta muda ao redor dos rios. Para isso, quantificamos a quantidade de floresta que ocorre em várias distâncias em pontos ao longo dos rios a montante das hidrelétricas no Rio Araguari. Para ilustrar esta abordagem geral, usamos o banco de dados MapBiomas Coleção 6 de 2020, e vincule esses dados de cobertura da terra aos pontos de amostragem em rios.

1.4.1 Obter e carregar dados (vectores)

Precisamos carregar os dados para rios e pontos de amostragem. Baixar arquivo (vector) com os dados (formato "GPKG", tamanho 54.9 MB). Este arquivo contém diferentes camadas vetoriais usadas para avaliar impactos de barragens hidroelétricas em tracajas ((Bárcenas-García et al. 2022)) e ariranhas ((Raffo et al. 2022)).

Mais sobre dados vetoriais. O formato aberto GeoPackage é um contêiner que permite armazenar dados SIG (feições/camadas) em um único arquivo. Por exemplo, um arquivo GeoPackage pode conter vários dados (dados vetoriais e raster) em diferentes sistemas de coordenadas. Todos esses recursos permitem que você compartilhe dados facilmente e evite a duplicação de arquivos.

Baixar o arquivo Link: https://github.com/darrennorris/gisdata/blob/master/inst/vector/rivers.gpkg . Lembrando-se de salvar o arquivo ("rivers.gpkg") em um local conhecido no seu computador.

O formato "GPKG" e diferente de "tif" (raster), o processo de importação é, portanto, diferente. Primeira, avisar R sobre onde ficar o arquivo. O código abaixo vai abrir uma nova janela, e você deve buscar e selecionar o arquivo "rivers.GPKG":

```
meuSIG <- file.choose()
```

Agora vamos olhar o que tem no arquivo. Depois que voces rodar o codigo **st_layers(meuSIG)** , o resultado mostra que o arquivo rivers.GPKG inclui camadas diferentes com pontos ("Point"), linhas ("Line String") e polígonos ("Polygon"). Alem disso, a coluna "crs_name" mostrar que a sistema de coordenadas é geográfica (WGS84, (EPSG: 4326)

$$https://epsg.io/4326$$

, e é diferente do arquivo raster:

```
sf::st_layers(meuSIG)
```

```
## Driver: GPKG
## Available layers:
##                   layer_name geometry_type features fields crs_name
## 1                  centerline   Line String       52     15   WGS 84
## 2                  forestloss         Point   276086     12   WGS 84
## 3                   canalpoly       Polygon        3      6   WGS 84
## 4              extentpoly50km       Polygon        1      0   WGS 84
## 5                   midpoints         Point       52     17   WGS 84
```

```
## 6            midpoints_hansen         Point       52     37   WGS 84
## 7         cachoeira_caldeirao         Point        1      2   WGS 84
## 8                porto_grande         Point        1      1   WGS 84
## 9                 icmbio_base         Point        1      1   WGS 84
## 10              direct_affect       Polygon        1      2   WGS 84
## 11 midpoints_hansen_distances         Point       52     43   WGS 84
## 12       midpoints_hansen_ffr         Point       52     82   WGS 84
## 13     midpoints_hansen_ffril         Point       52     91   WGS 84
## 14         direct_affect_line   Line String        1      2   WGS 84
```

Nós só precisamos de duas dessas camadas. O código abaixo vai carregar as camadas que precisamos e criar os objetos "rsm" e "rsl". Assim, agora temos dados com: pontos cada 5 km ao longo os rios ("rsm") e a linha central de rios ("rsl").

```
# pontos cada 5 km
rsm <- sf::st_read(meuSIG, layer = "midpoints")
```

```
## Reading layer `midpoints' from data source
##   `C:\Users\user\Documents\Articles\gis_layers\gisdata\inst\vector\rivers.gpkg'
##   using driver `GPKG'
## Simple feature collection with 52 features and 17 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -52.01259 ymin: 0.7175827 xmax: -51.29688 ymax: 1.330365
## Geodetic CRS:  WGS 84
```

```
# linha central de rios
rsl <- sf::st_read(meuSIG, layer = "centerline")
```

```
## Reading layer `centerline' from data source
##   `C:\Users\user\Documents\Articles\gis_layers\gisdata\inst\vector\rivers.gpkg'
##   using driver `GPKG'
## Simple feature collection with 52 features and 15 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: -52.01443 ymin: 0.7094595 xmax: -51.2924 ymax: 1.352094
## Geodetic CRS:  WGS 84
```

1.4.2 Visualizar os arquivos (camadas vector)

Visualizar para verificar. Mapa com linha central e pontos de rios em trechos de 5km.

```
ggplot(rsl) +
  geom_sf(aes(color=rio)) +
  geom_sf(data = rsm, shape=21, aes(fill=zone))
```

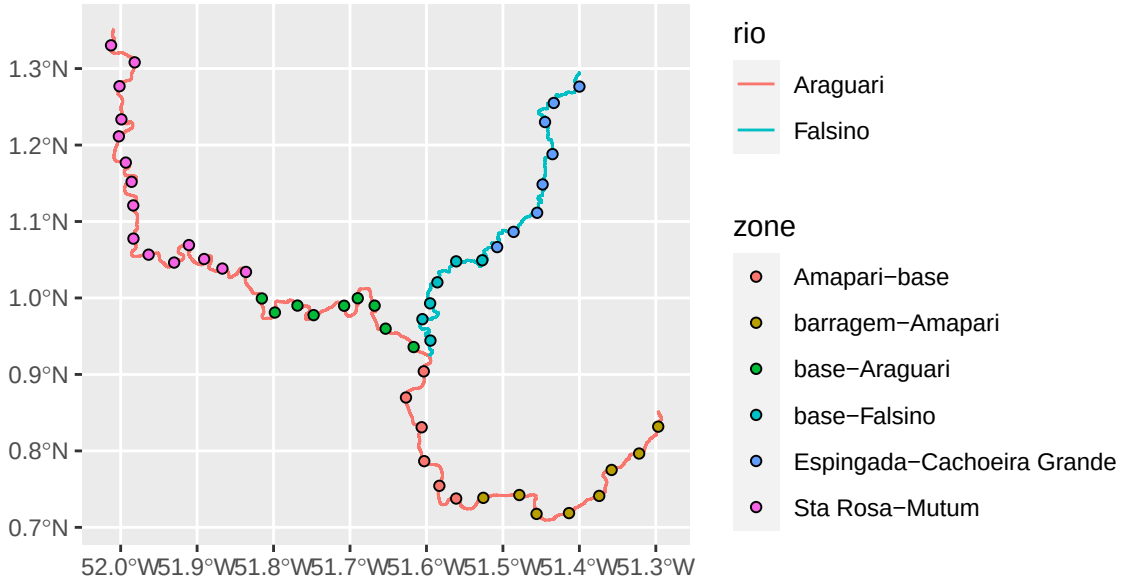

Figure 7: Pontos ao longo dos rios a montante das hidrelétricas no Rio Araguari.

Mapa interativo (funcione somente com internet) Mostrando agora com fundo de mapas "base" (OpenStreetMap/ESRI etc)

```
#
mapview(rsl, zcol = "rio")
```

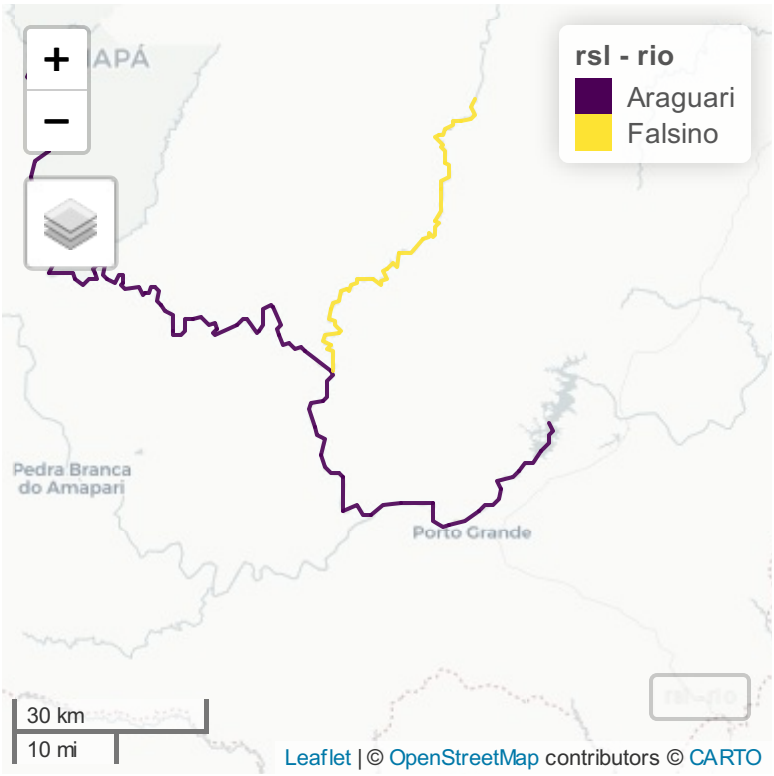

Figure 8: Linhas dos rios a montante das hidrelétricas no Rio Araguari.

1.4.3 Obter e carregar dados (raster)

Mais uma vez vamos aproveitar os dados de MapBiomas. Agora baixar arquivo raster com cobertura de terra no entorno dos rios em 2020, (formato ".tif", tamanho 3.3 MB). Link: https://github.com/darrennorris/gisdata/blob/master/inst/raster/mapbiomas_AP_utm_rio/utm_cover_AP_rio_2020.tif . Lembrando-se de salvar o arquivo ("utm_cover_AP_rio_2020.tif") em um local conhecido no seu computador. Agora avisar R sobre onde ficar o arquivo. O código abaixo vai abrir uma nova janela, e você deve buscar e selecionar o arquivo "utm_cover_AP_rio_2020.tif":

```
meuSIGr <- file.choose()
```

O código abaixo vai carregar os dados e criar o objeto "mapbiomas_2020".

```
mapbiomas_2020 <- rast(meuSIGr)
```

1.4.4 Visualizar os arquivos (camadas raster e vector)

Visualizar para verificar. É possível de visualizar camadas de raster e vetor juntos com funcões no pacote Tmap (https://r-tmap.github.io/tmap-book/index.html).

```
# Passo necessario para agilizar o processamento
mapbiomas_2020_modal<-aggregate(mapbiomas_2020, fact=10, fun="modal")
# Plot
tm_shape(mapbiomas_2020_modal) +
  tm_raster(title = "Classe", style = "cat", palette = "Set3") +
tm_shape(rsl) +
  tm_lines(col="blue") +
tm_shape(rsm) +
  tm_dots(size = 0.2, col = "yellow") +
```

```
tm_compass(position=c("left", "top")) +
tm_scale_bar(breaks = c(0, 25, 50), text.size = 1,
             position=c("left", "bottom")) +
tm_layout(legend.position = c("right","top"), legend.bg.color="white")
```

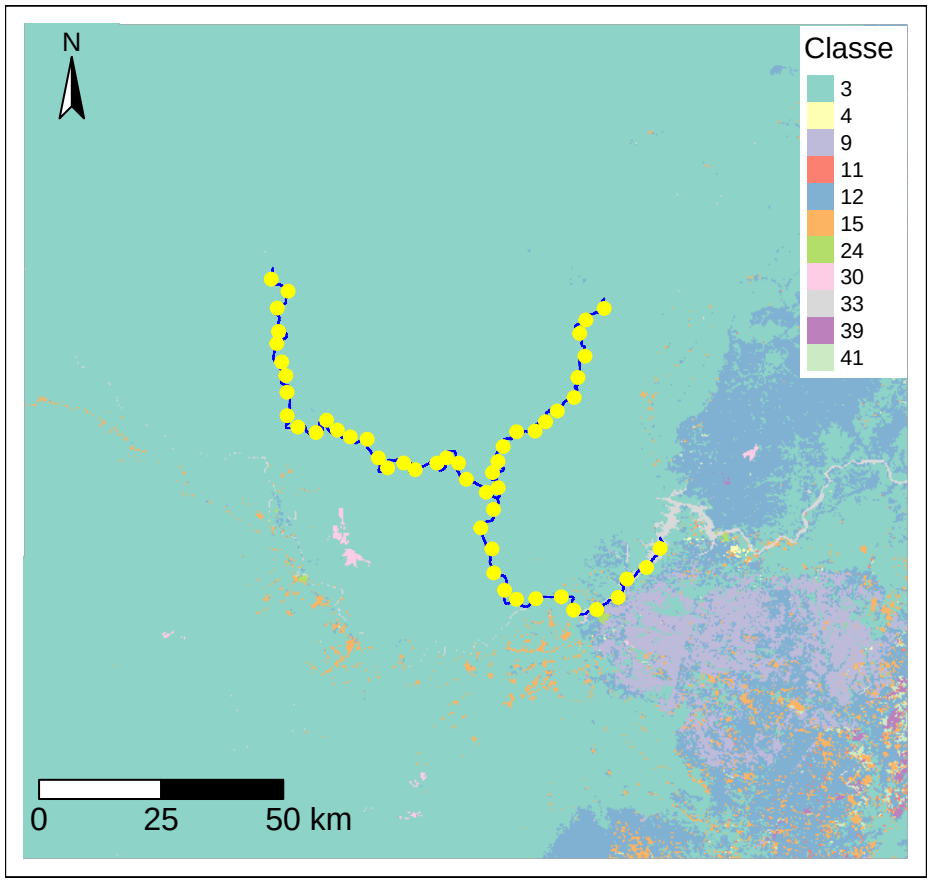

Figure 9: Cobertura da terra ao redor do Rio Araguari em 2020. Mostrando os pontos de amostragem (pontos amarelas) cada 5 quilômetros ao longo do rio.

1.4.5 Reclassificação

Para simplificar nossa avaliação de escala, reclassificamos a camada mapbiomas_2020 em uma camada binária de floresta/não-floresta. Essa tarefa de geoprocessamento pode ser realizada anteriormente usando SIG (QGIS). Aqui vamos reclassificar as categorias de cobertura da terra (agrupando diferentes áreas de cobertura florestal tipos) usando alguns comandos genéricos do R para criar uma nova camada com a cobertura de floresta em toda a região de estudo. Para isso, criamos um mapa do mesmo resolução e extensão, e então podemos redefinir os valores do mapa. Neste caso, queremos agrupar a cobertura da terra categorias 3 e 4 (Formação Florestal e Formação Savânica, respectivamente).

```
# criar uma nova camada de floresta
floresta_2020 <- mapbiomas_2020
# Com valor de 0
values(floresta_2020) <- 0
# Atualizar categorias florestais agrupados com valor de 1
floresta_2020[mapbiomas_2020==3 | mapbiomas_2020==4] <- 1
```

Vizualizar para verificar.

```
# Passo necessario para agilizar o processamento
floresta_2020_modal<-aggregate(floresta_2020, fact=10, fun="modal")
# Plot
tm_shape(floresta_2020_modal) +
  tm_raster(style = "cat",
            palette = c("0" = "#E974ED", "1" ="#129912"), legend.show = FALSE) +
  tm_add_legend(type = "fill", labels = c("não-floresta", "floresta"),
    col = c("#E974ED", "#129912"), title = "Classe") +
tm_shape(rsl) +
  tm_lines(col="blue") +
tm_shape(rsm) +
  tm_dots(size = 0.2, col = "yellow") +
tm_scale_bar(breaks = c(0, 25, 50), text.size = 1,
             text.color = "white", position=c("left", "bottom")) +
tm_layout(legend.position = c("right","top"),legend.bg.color = "white")
```

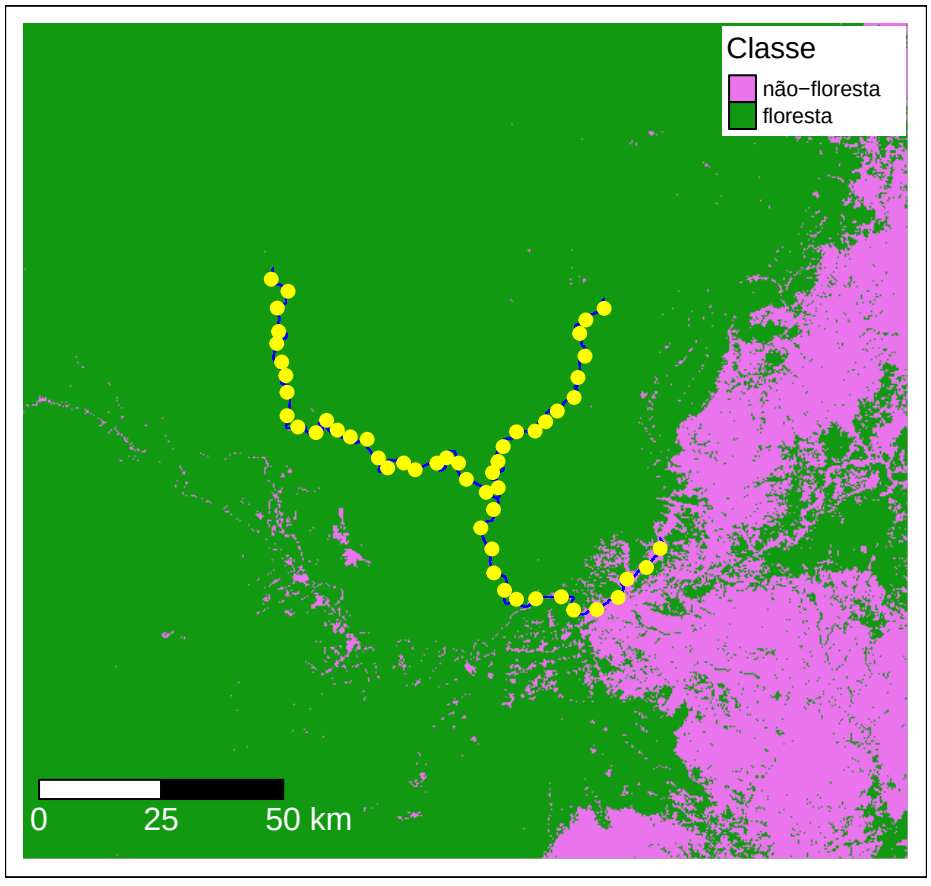

Figure 10: Floresta ao redor do Rio Araguari. MapBiomas 2020 reclassificado em floresta e não-floresta. Mostrando os pontos de amostragem (pontos amarelas) cada 5 quilômetros ao longo do rio.

1.5 Comparação multiescala

Em seguida, com as coordenadas dos pontos das localizações das amostras calculamos a quantidade de floresta que circunda cada local de amostragem em diferentes extensões.

```
rsm_31976 <- st_transform(rsm, 31976)
# Buffer
rsm_31976_b1000 <- st_buffer(rsm_31976[1, ], dist = 1000)

# Recorte com buffer de 1000 metros (mudando a extensão).
buffer.forest1.1km <- crop(floresta_2020, snap="out", rsm_31976_b1000)
# Máscara para que os pixels fora do polígono sejam nulos.
buffer.forest1.1km <- mask(buffer.forest1.1km, rsm_31976_b1000, touches=TRUE)
names(buffer.forest1.1km) <- "forest_2020_1km"
```

Vizualizar para verificar.

```
# Plot
tm_shape(buffer.forest1.1km) +
  tm_raster(style = "cat",
            palette = c("0" = "#E974ED",
                        "1" ="#129912"), legend.show = FALSE) +
tm_shape(rsm_31976[1, ]) +
  tm_symbols(shape =21, col = "yellow",
             border.col = "black", border.lwd = 0.2, size=0.5) +
tm_shape(rsm_31976_b1000) +
  tm_borders(col = "black", lwd = 4, lty = "dashed") +
tm_add_legend(type = "fill",  labels = c("não-floresta", "floresta"),
    col = c("#E974ED", "#129912"), title = "Classe") +
tm_compass(position=c("left", "top")) +
tm_scale_bar(breaks = c(0, 0.5, 1), text.size = 1,
             position=c("left", "bottom")) +
tm_layout(legend.position = c("right","top"), legend.bg.color = "white")
```

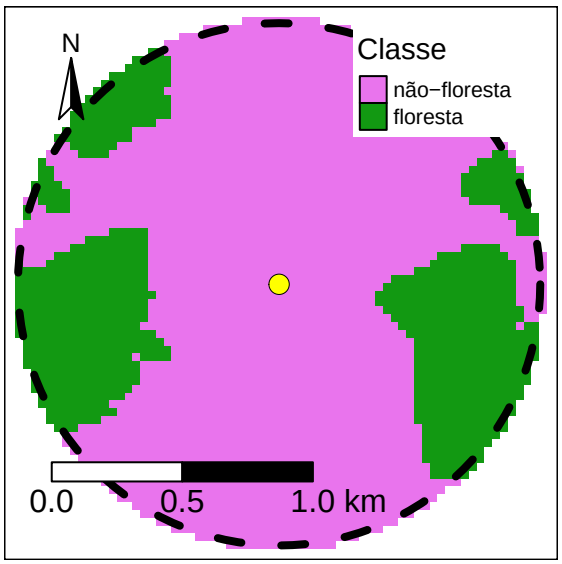

Figure 11: Ilustração da determinação da quantidade de habitat ao redor de um ponto. Para um determinada extensão, o habitat de interesse é isolado. Um buffer (linha tracejada) é colocado ao redor de um ponto (amarela) e o número de células (pixels) que contém o habitat é somado e multiplicado pela área de cada pixel.

1.5.1 Pergunta 3

Qual é a extensão em número de pixels desse recorte (buffer.forest1.1km)?

Temos valores de 0 (não-floresta) e 1 (floresta). Então, para saber a aréa de floresta podemos somar o número de células (pixels) que contém o habitat e multiplica pela área de cada pixel conforme o codigo:

```
# 1) Somatório.
# No caso igual o numero de pixels de floresta.
# Para todo a paisagem, somatorio "global".
# Não deve incluir pixels nulos, então use "na.rm = TRUE".
```

```r
soma_floresta <- global(buffer.forest1.1km, "sum", na.rm = TRUE)
soma_floresta
```

```
##                  sum
## forest_2020_1km 943
```

```r
# 2) Área de cada pixel.
# Sabemos o sistema de coordenadas (EPSG = 31976).
# EPSG 31976 é uma sistema projetado com unidade em metros.
buffer.forest1.1km
```

```
## class       : SpatRaster
## dimensions  : 68, 68, 1  (nrow, ncol, nlyr)
## resolution  : 29.89281, 29.89281  (x, y)
## extent      : 465959.3, 467992, 90921.47, 92954.18  (xmin, xmax, ymin, ymax)
## coord. ref. : SIRGAS 2000 / UTM zone 22N (EPSG:31976)
## source(s)   : memory
## name        : forest_2020_1km
## min value   :               0
## max value   :               1
```

```r
# Portanto, o tamanho de cada pixel é igual.
area_pixel_m2 <- 29.89281 * 29.89281
area_pixel_m2
```

```
## [1] 893.5801
```

```r
# 3) Calculos de aréa.
# Aréa de floresta m2
area_floresta_m2 <- soma_floresta * area_pixel_m2
area_floresta_m2
```

```
##                    sum
## forest_2020_1km 842646
```

```r
# Aréa de floresta hectares
area_floresta_ha <- area_floresta_m2 / 10000
area_floresta_ha
```

```
##                     sum
## forest_2020_1km 84.2646
```

Para uma comparação multiescala, vamos repetir o mesmo processo, mas agora com distancias de 250, 500, 1000, 2000 e 4000 metros, doubrando a escala (extensão) em cada passo.

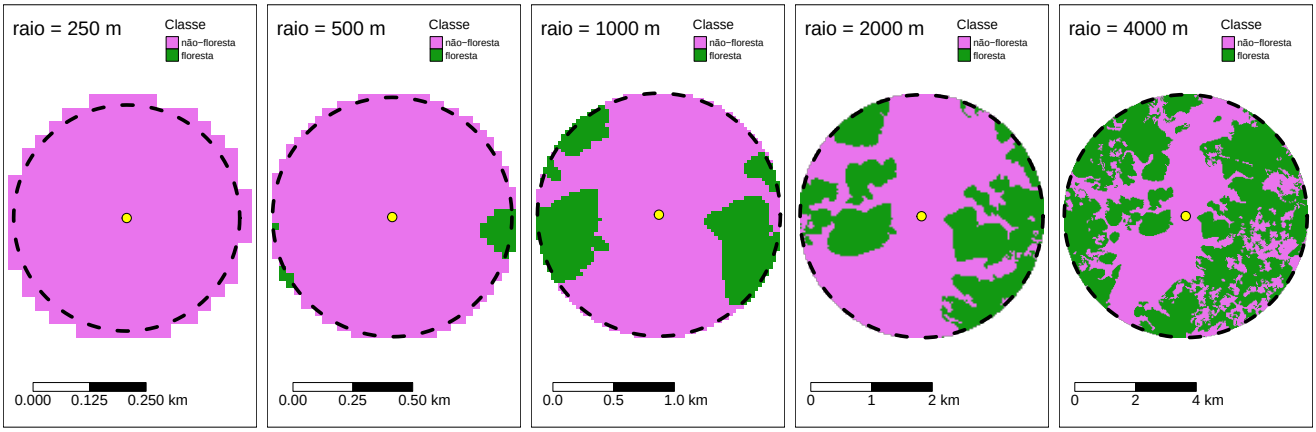

Figure 12: Cobertura florestal em extensões diferentes ao redor de um local de amostragem.

Aspectos quantitativos das paisagens mudam fundamentalmente com a escala. Por exemplo, nesse caso, parece que a proporção de floresta aumenta à medida que a extensão aumenta de 500 para 4000 metros. Esta percepção visual é confirmada pelos valores calculados, onde as áreas são:

- raio 250 m = 0 hectares de floresta
- raio 500 m = 6,3 hectares de floresta
- raio 1000 m = 84,3 hectares de floresta
- raio 2000 m = 502.6 hectares de floresta
- raio 4000 m = 3351.0 hectares de floresta

1.5.2 Pergunta 4

Usando os valores listadas acima de raio e área de floresta para os diferentes buffers circulares, calcule a proporção de floresta em cada uma das diferentes extensões de buffer. Apresente 1) os resultados incluindo cálculos. 2) um gráfico com valores de extensão no eixo x e proporção da floresta no eixo y. 3) Em menos de 200 palavras apresente a sua interpretação do gráfico.

1.5.3 Pergunta 5

A modelagem multiescala quantifica as condições do ambiente em múltiplas escalas alterando o resolução ou a extensão da análise e, em seguida, avaliando qual das escalas consideradas explica melhor um padrão ou processo. Escolha 1 espécie aquático e 1 espécie terrestre que ocorram na região a montante das hidrelétricas no Rio Araguari. Com base nas diferenças entre extensões (indicados no exemplo anterior) e as características funcionais das espécies (por exemplo área de vida), escolher as extensões mais adequadas para um estudo multiescala de cada espécie.

1.6 Próximos passos: repetindo para muitas amostras.

Neste exemplo comparamos a área de floresta em torno de um único ponto de amostragem. Para calcular o mesmo para todos os 52 pontos, seriam necessárias varias repetições (52 pontos x 5 extensões = 260 repetições).

Poderíamos escrever código para executar esse processo automaticamente. Felizmente, alguém já escreveu funções para fazer isso e muito mais. O próximo tutorial sobre métricas de paisagem mostrará exemplos usando o pacote "landscapemetrics" (https://r-spatialecology.github.io/landscapemetrics/).

2 Codigo no livro

- Objetivo não é de apresentar detalhes sobre os cálculos/métodos estatísticas ou os funções no R. Existem diversos exemplos disponíveis "Ciência de Dados com R–Introdução.....": e com google "r cran introdução tutorial"..... Alem disso, existem grupos de ajuda, como por exemplo: R Brasil e Stack Overflow em Português
- O objetivo é de apresentar um livro mostrado os capacidades e opções para desenvolver e integrar pesquisas na ecologia da paisagem no ambiente estatística de R

Obviamente, todas as tarefas de geoprocessamento podem ser desenvolvidas anteriormente em um SIG (QGIS). Então porque use R? R tem a capacidade (baseada em codigo) para alternar entre tarefas de processamento, modelagem e visualização de dados geográficos e não geográficos. Alem disso, como é possível importar, modificar, analisar e visualizar dados espaciais no mesmo ambiente com script/codigo, o R permite fluxos de trabalho transparentes e reproduzíveis (A Ciência Aberta).

Aliás, atualmente a grande maioria dos artigos científicos publicados na revista Landscape Ecology incluir análises usando R.

2.1 Organização do codigo no livro

O capitulo está organizado em etapas de processamento, com blocos de código em caixas cinzas:

```
codigo de R para executar
```

Para segue os passos, os blocos de código precisam ser executados em sequência. Se você pular uma etapa, ou rodar fora de sequência o próximo bloco de código provavelmente não funcionará.

As linhas de codigo de R dentro de cada caixa tambem preciso ser executados em sequência. O simbolo # é usado para incluir comentarios sobre os passos no codgio (ou seja, linhas começando com # não é codigo de executar).

```
# Passo 1
codigo de R passo 1 # texto e numeros tem cores diferentes
# Passo 2
codigo de R passo 2
# Passo 3
codigo de R passo 3
```

Alem disso, os simbolos #> e/ou [**1**] no início de uma linha indica o resultado que você verá no console de R.

```
# Passo 1
1+1
```

[1] 2

```
# Passo 2
x <- 1 + 1
# Passo 3
x
```

[1] 2

```
# Passo 4
x + 1
```

[1] 3

3 Codigo no livro

- Objetivo não é de apresentar detalhes sobre os cálculos/métodos estatísticas ou os funções no R. Existem diversos exemplos disponíveis "Ciência de Dados com R–Introdução.....": e com google "r cran introdução tutorial"..... Alem disso, existem grupos de ajuda, como por exemplo: R Brasil e Stack Overflow em Português
- O objetivo é de apresentar um livro mostrado os capacidades e opções para desenvolver e integrar pesquisas na ecologia da paisagem no ambiente estatística de R

Obviamente, todas as tarefas de geoprocessamento podem ser desenvolvidas anteriormente em um SIG (QGIS). Então porque use R? R tem a capacidade (baseada em codigo) para alternar entre tarefas de processamento, modelagem e visualização de dados geográficos e não geográficos. Alem disso, como é possível importar, modificar, analisar e visualizar dados espaciais no mesmo ambiente com script/codigo, o R permite fluxos de trabalho transparentes e reproduzíveis (A Ciência Aberta).

Aliás, atualmente a grande maioria dos artigos científicos publicados na revista Landscape Ecology incluir análises usando R.

3.1 Organização do codigo no livro

O capitulo está organizado em etapas de processamento, com blocos de código em caixas cinzas:

```
codigo de R para executar
```

Para segue os passos, os blocos de código precisam ser executados em sequência. Se você pular uma etapa, ou rodar fora de sequência o próximo bloco de código provavelmente não funcionará.

As linhas de codigo de R dentro de cada caixa tambem preciso ser executados em sequência. O simbolo # é usado para incluir comentarios sobre os passos no codgio (ou seja, linhas começando com # não é codigo de executar).

```
# Passo 1
codigo de R passo 1 # texto e numeros tem cores diferentes
# Passo 2
codigo de R passo 2
# Passo 3
codigo de R passo 3
```

Alem disso, os simbolos #> e/ou [**1**] no início de uma linha indica o resultado que você verá no console de R.

```
# Passo 1
1+1
```

[1] 2

```
# Passo 2
x <- 1 + 1
# Passo 3
x
```

[1] 2

```
# Passo 4
x + 1
```

[1] 3

Codigo e R

Objetivo não é de apresentar grandes detalhes sobre os cálculos/métodos estatísticas ou os funções no R.

4 Primeiros passos com uma raster

Uma raster é um matriz de valores com coordenadas geográficos. Cada pixel de uma raster representa uma região geográfica, e o valor do pixel representa uma característica dessa região (mais sobre dados raster).

Em geral é necessário baixar alguns pacotes para que possamos fazer as nossas análises. Precisamos os seguintes pacotes, que deve esta instalado antes:

- tidyverse,
- terra,
- sf,
- mapview,
- tmap.

Carregar pacotes:

```
library(tidyverse)
library(terra)
library(sf)
library(mapview)
library(tmap)
```

Inicialmente iremos gerar uma raster representando uma paisagem bem simples, de 6 por 6 pixels. Você já deve saber que pixel é a unidade básica de uma imagem (lembra da camera do seu celular, 10Mb ou algo assim?!). Vocês devem ter visto sobre pixels e resolução no mesmo em aulas de geoprocessamento. Aqui podemos tratar o pixel como a resolução. Vamos dizer que temos um pixel de 10 metros (res=10 no bloco de código), ou seja, uma quadrado de 10 por 10m, sendo essa, a menor unidade mapeável. Assim sendo, a resolução tambem tem ligação com escala cartográfica!

Vamos gerar e plotar uma paisagem simples em 4 passos. Primeiramente, a função `rast` cria um objeto do tipo raster. E depois a função `values` atribui valores, e na sequencia vamos visualisar os valores com `plot` e `text` .

```
#Função "rast" gera a paisagem virtual (paisagem simulado)
pai_sim <- rast(ncols=6, nrows=6,
                xmin=1, xmax=60,
                ymin=1, ymax=60,
                res=10)
#E essa atribui valores ("values") para os pixels criados acima
values(pai_sim) <- 1
plot(pai_sim) #Essa plota
text(pai_sim) #Essa coloca os valores dos pixels
```

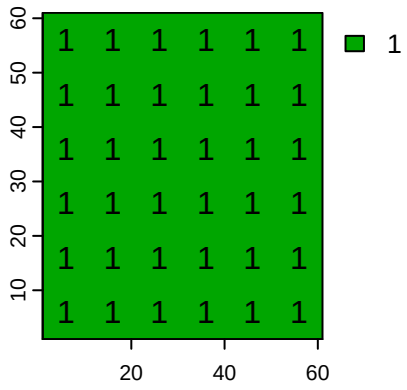

Figure 13: Paisagem simples de 36 pixels.

4.0.1 Obter e carregar dados (raster)

Mais uma vez vamos aproveitar os dados de MapBiomas. Agora baixar arquivo raster com cobertura de terra no entorno dos rios em 2020, (formato ".tif", tamanho 3.3 MB). Link: https://github.com/darrennorris/gisdata/blob/master/inst/raster/mapbiomas_AP_utm_rio/utm_cover_AP_rio_2020.tif . Lembrando-se de salvar o arquivo ("utm_cover_AP_rio_2020.tif") em um local conhecido no seu computador. Agora avisar R sobre onde ficar o arquivo. O código abaixo vai abrir uma nova janela, e você deve buscar e selecionar o arquivo "utm_cover_AP_rio_2020.tif":

```
meuSIGr <- file.choose()
```

O código abaixo vai carregar os dados e criar o objeto “mapbiomas_2020”.

```
mapbiomas_2020 <- rast(meuSIGr)
```

4.0.2 Reclassificação

Para simplificar nossa avaliação de escala, reclassificamos a camada mapbiomas_2020 em uma camada binária de floresta/não-floresta. Essa tarefa de geoprocessamento pode ser realizada anteriormente usando SIG (QGIS). Aqui vamos reclassificar as categorias de cobertura da terra (agrupando diferentes áreas de cobertura florestal tipos) usando alguns comandos genéricos do R para criar uma nova camada com a cobertura de floresta em toda a região de estudo. Para isso, criamos um mapa do mesmo resolução e extensão, e então podemos redefinir os valores do mapa. Neste caso, queremos agrupar a cobertura da terra categorias 3 e 4 (Formação Florestal e Formação Savânica, respectivamente).

```
# criar uma nova camada de floresta
floresta_2020 <- mapbiomas_2020
# Com valor de 0
values(floresta_2020) <- 0
# Atualizar categorias florestais agrupados com valor de 1
floresta_2020[mapbiomas_2020==3 | mapbiomas_2020==4] <- 1
```

Vizualizar para verificar.

```
# Passo necessario para agilizar o processamento
floresta_2020_modal<-aggregate(floresta_2020, fact=10, fun="modal")
# Plot
tm_shape(floresta_2020_modal) +
  tm_raster(style = "cat",
            palette = c("0" = "#E974ED", "1" ="#129912"),
            legend.show = FALSE) +
  tm_add_legend(type = "fill", labels = c("não-floresta", "floresta"),
    col = c("#E974ED", "#129912"), title = "Classe") +
tm_scale_bar(breaks = c(0, 25, 50), text.size = 1,
             text.color = "white", position=c("left", "bottom")) +
tm_layout(legend.position = c("right","top"),legend.bg.color = "white")
```

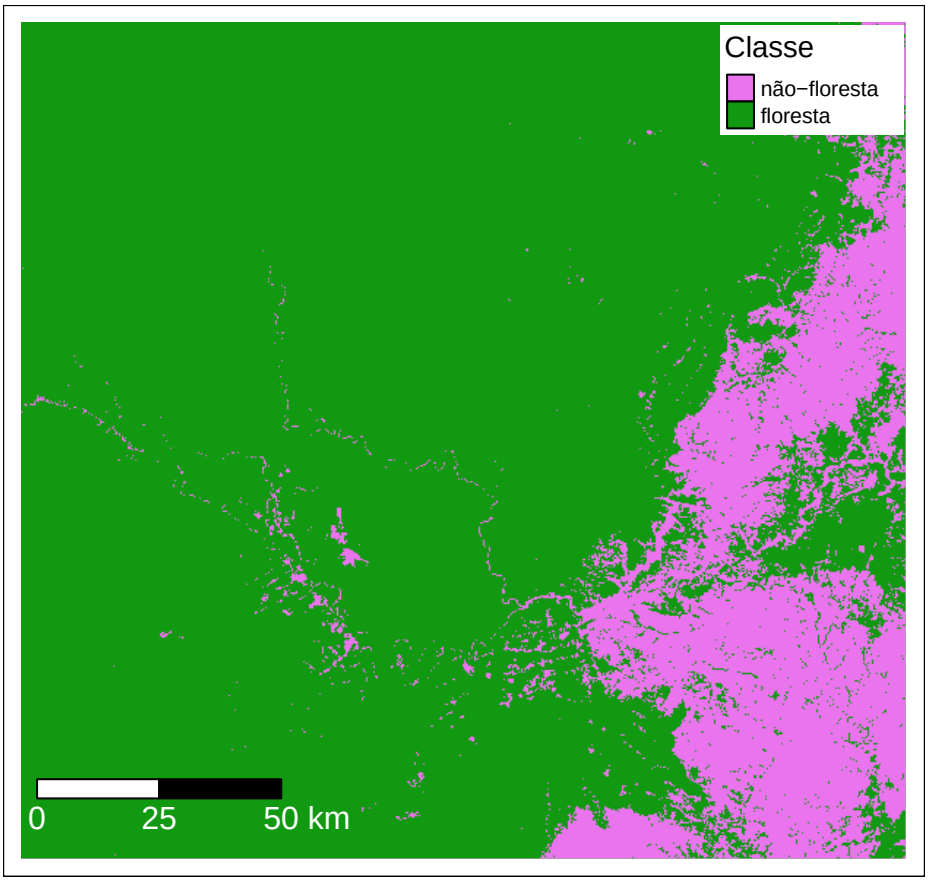

Figure 14: Floresta ao redor do Rio Araguari. MapBiomas 2020 reclassificado em floresta e não-floresta. Mostrando os pontos de amostragem (pontos amarelas) cada 5 quilômetros ao longo do rio.

5 Primeiros passos com vector

Em geral é necessário baixar alguns pacotes para que possamos fazer as nossas análises. Precisamos os seguintes pacotes, que deve esta instalado antes:

- tidyverse,
- sf,
- mapview,
- tmap.

Carregar pacotes:

```
library(tidyverse)
library(sf)
library(mapview)
library(tmap)
```

5.1 Obter e carregar dados (vectores)

Precisamos carregar os dados para rios e pontos de amostragem. Baixar arquivo (vector) com os dados (formato "GPKG", tamanho 54.9 MB). Mais sobre dados vetoriais. O formato aberto GeoPackage é um contêiner que permite armazenar dados SIG (feições/camadas) em um único arquivo. Por exemplo, um arquivo GeoPackage pode conter vários dados (dados vetoriais e raster) em diferentes sistemas de coordenadas. Todos esses recursos permitem que você compartilhe dados facilmente e evite a duplicação de arquivos.

Baixar o arquivo Link: https://github.com/darrennorris/gisdata/blob/master/inst/vector/rivers.gpkg . Lembrando-se de salvar o arquivo ("rivers.gpkg") em um local conhecido no seu computador.

O formato "GPKG" e diferente de "tif" (raster), o processo de importação é, portanto, diferente. Primeira, avisar R sobre onde ficar o arquivo. O código abaixo vai abrir uma nova janela, e você deve buscar e selecionar o arquivo "rivers.GPKG":

```
meuSIG <- file.choose()
```

Agora vamos olhar o que tem no arquivo. Depois que voces rodar o codigo **st_layers(meuSIG)** , o resultado mostra que o arquivo rivers.GPKG inclui camadas diferentes com pontos ("Point"), linhas ("Line String") e polígonos ("Polygon"). Alem disso, a coluna "crs_name" mostrar que a sistema de coordenadas é geográfica (WGS84, (EPSG: 4326)

$$https://epsg.io/4326$$

, e é diferente do arquivo raster:

```
sf::st_layers(meuSIG)
```

```
## Driver: GPKG
## Available layers:
##                    layer_name geometry_type features fields crs_name
## 1                  centerline   Line String       52     15   WGS 84
## 2                  forestloss         Point   276086     12   WGS 84
## 3                   canalpoly       Polygon        3      6   WGS 84
## 4               extentpoly50km       Polygon        1      0   WGS 84
## 5                   midpoints         Point       52     17   WGS 84
## 6            midpoints_hansen         Point       52     37   WGS 84
## 7         cachoeira_caldeirao         Point        1      2   WGS 84
## 8                porto_grande         Point        1      1   WGS 84
## 9                 icmbio_base         Point        1      1   WGS 84
## 10              direct_affect       Polygon        1      2   WGS 84
## 11 midpoints_hansen_distances         Point       52     43   WGS 84
## 12       midpoints_hansen_ffr         Point       52     82   WGS 84
## 13     midpoints_hansen_ffril         Point       52     91   WGS 84
## 14         direct_affect_line   Line String        1      2   WGS 84
```

Nós só precisamos de duas dessas camadas. O código abaixo vai carregar as camadas que precisamos e criar os objetos "rsm" e "rsl". Assim, agora temos dados com: pontos cada 5 km ao longo os rios ("rsm") e a linha central de rios ("rsl").

```
# pontos cada 5 km
rsm <- sf::st_read(meuSIG, layer = "midpoints")
```

```
## Reading layer `midpoints' from data source
##   `C:\Users\user\Documents\Articles\gis_layers\gisdata\inst\vector\rivers.gpkg'
##   using driver `GPKG'
## Simple feature collection with 52 features and 17 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: -52.01259 ymin: 0.7175827 xmax: -51.29688 ymax: 1.330365
## Geodetic CRS:  WGS 84
```

```
# linha central de rios
rsl <- sf::st_read(meuSIG, layer = "centerline")
```

```
## Reading layer `centerline' from data source
##   `C:\Users\user\Documents\Articles\gis_layers\gisdata\inst\vector\rivers.gpkg'
##   using driver `GPKG'
## Simple feature collection with 52 features and 15 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: -52.01443 ymin: 0.7094595 xmax: -51.2924 ymax: 1.352094
## Geodetic CRS:  WGS 84
```

5.2 Visualizar os arquivos (camadas vector)

Visualizar para verificar. Mapa com linha central e pontos de rios em trechos de 5km.

```
ggplot(rsl) +
  geom_sf(aes(color=rio)) +
  geom_sf(data = rsm, shape=21, aes(fill=zone))
```

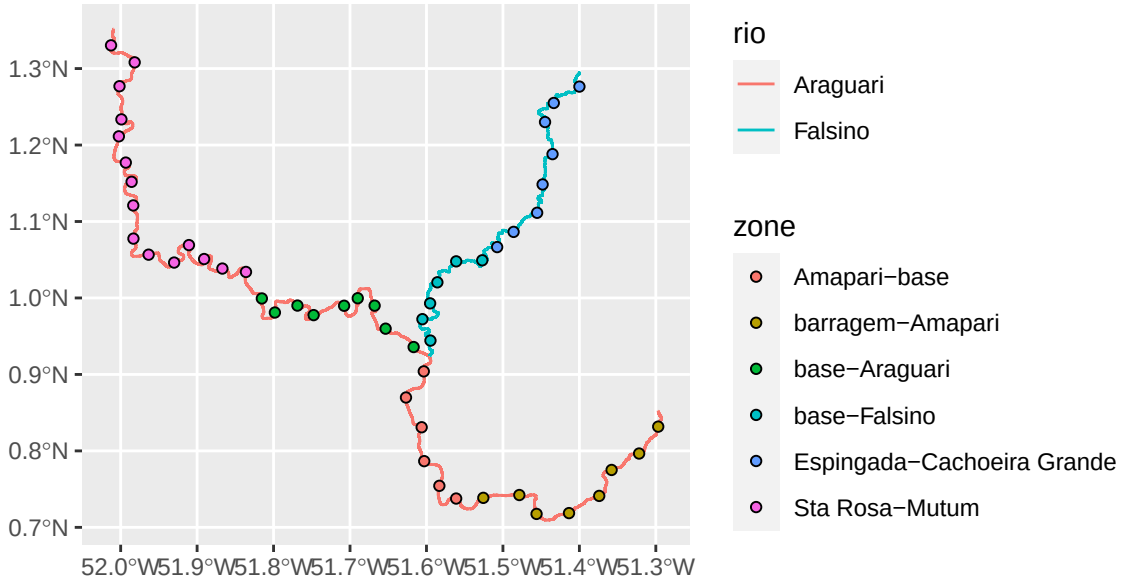

Figure 15: Pontos ao longo dos rios a montante das hidrelétricas no Rio Araguari.

Mapa interativo (funcione somente com internet) Mostrando agora com fundo de mapas "base" (OpenStreetMap/ESRI etc)

```
#
mapview(rsl, zcol = "rio")
```

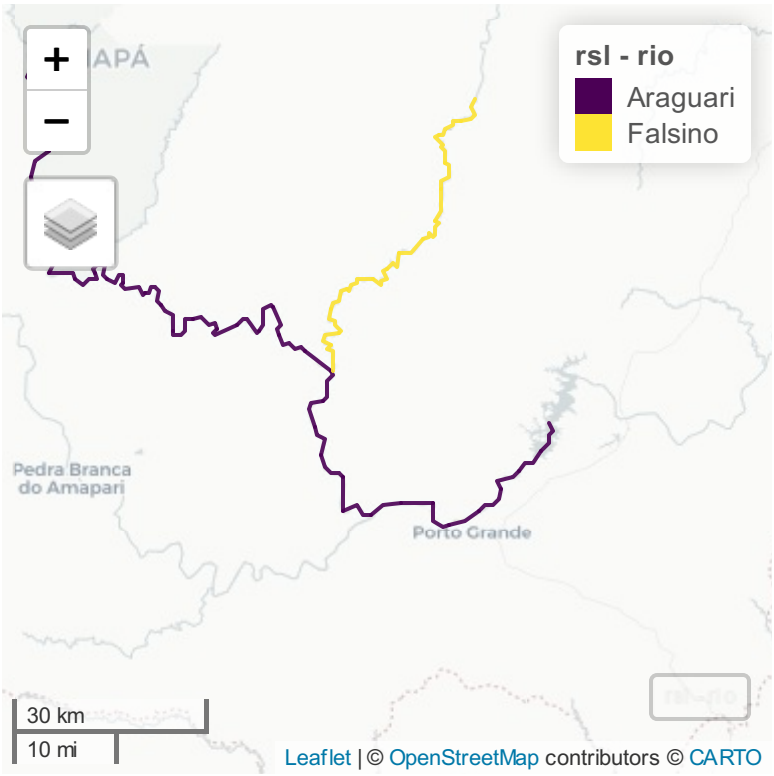

Figure 16: Linhas dos rios a montante das hidrelétricas no Rio Araguari.

Bárcenas-García, Andrea, Fernanda Michalski, James P. Gibbs, and Darren Norris. 2022. "Amazonian Run-of-River Dam Reservoir Impacts Underestimated: Evidence from a Before–after Control–impact Study of Freshwater Turtle Nesting Areas." *Aquatic Conservation: Marine and Freshwater Ecosystems* 32 (3): 508–22. https://doi.org/10.1002/aqc.3775.

Fletcher, Robert, and Marie-Josée Fortin. 2018. *Spatial Ecology and Conservation Modeling*. Springer International Publishing. https://doi.org/10.1007/978-3-030-01989-1.

Raffo, Deborah C. Dávila, Darren Norris, Sandra Maria Hartz, and Fernanda Michalski. 2022. "Anthropogenic Influences on the Distribution of a Threatened Apex-Predator Around Sustainable-Use Reserves Following Hydropower Dam Installation." *PeerJ* 10 (October): e14287. https://doi.org/10.7717/peerj.14287.