

Ecologia de Paisagens com R: soluções

Darren Norris

2024-01-03

Sumário

Bem-vindos	4
I Apresentação	5
Introdução	6
Organização do código no livro	6
Running R code	7
II Soluções	9
1 Capítulo Escala	10
1.1 Pergunta 1	10
1.1.1 resposta	10
1.2 Pergunta 2	11
1.2.1 resposta	11
1.3 Pergunta 3	12
1.3.1 resposta	12
1.4 Pergunta 4	12
1.4.1 resposta	12
1.5 Pergunta 5	13
1.5.1 resposta	13
1.6 Pergunta 6	15
1.6.1 resposta	15
2 Capítulo Métricas	16
2.1 Pergunta 1	16
2.1.1 resposta	16
2.2 Pergunta 2	17
2.2.1 resposta	17
2.3 Pergunta 3	17
2.3.1 resposta	18
2.4 Pergunta 4	19
2.4.1 resposta	19

2.5	Pergunta 5	19
2.5.1	resposta	20
2.6	Pergunta 6	20
2.6.1	resposta	20
2.7	Pergunta 7	23
2.7.1	resposta	24
2.8	Pergunta 8	24
2.8.1	resposta	24
References		25

Bem-vindos

Este livro inclui soluções para perguntas do livro *Ecologia de Paisagens com R* : <https://darrennorris.github.io/epr/>.

Parte I

Apresentação

Introdução

O foco aqui é fornecer soluções para as questões em que você precisa executar o código R. Para as questões que requerem interpretação de figuras e compreensão das informações fornecidas nos capítulos serão fornecidas algumas dicas para ajudar e orientar as respostas individuais.

Organização do código no livro

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

```
codigo de R para executar
```

Para seguir 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 código de R dentro de cada caixa também precisam ser executados em sequência. O símbolo `#` é usado para incluir comentários sobre os passos no código (ou seja, linhas começando com `#` não é código 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 símbolos `#>` 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
```

```
1 + 1
```

```
[1] 2
```

Examples from <https://github.com/hadley/r4ds/blob/main/intro.qmd>

You can install the complete tidyverse with a single line of code:

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

On your computer, type that line of code in the console, and then press enter to run it. R will download the packages from CRAN and install them on your computer.

Running R code

The previous section showed you examples of running R code. The code in the book looks like this:

```
1 + 2
```

```
[1] 3
```

If you run the same code in your local console, it will look like this:

```
> 1 + 2
[1] 3
```

There are two main differences. In your console, you type after the `>`, called the **prompt**; we don't show the prompt in the book. In the book, the output is commented out with `#>`; in your console, it appears directly after your code. These two differences mean that if you're working with an electronic version of the book, you can easily copy code out of the book and paste it into the console.

Throughout the book, we use a consistent set of conventions to refer to code:

- Functions are displayed in a code font and followed by parentheses, like `sum()` or `mean()`.
- Other R objects (such as data or function arguments) are in a code font, without parentheses, like `flights` or `x`.
- Sometimes, to make it clear which package an object comes from, we'll use the package name followed by two colons, like `dplyr::mutate()` or `nycflights13::flights`. This is also valid R code.

Parte II

Soluções

1 Capítulo Escala

Carregar os pacotes necessarios:

```
library(tidyverse)
library(sf)
library(terra)
library(eprdados)
```

1.1 Pergunta 1

Sobre o objeto ramostra. Com base nos resultados obtidos, qual o área do pixel em metros quadrados? Qual o área total da paisagem em hectares e quilometros quadrados?

1.1.1 resposta

Para responder a pergunta você deve revisar o conteúdo no capítulo Escala, especificamente:

1. A figura mostrando os “componentes de uma raster e suas atributos no pacote terra”.
2. Os exemplos de código mostrando “como obter informações específicas através de funções específicas”.

Sabemos que a sistema de coordenados é “projetado”. Assim sendo, todos os pixels no objeto tem as mesmas dimensões. Portanto, o calculo é mais simples em comparação com o caso onde a sistema de coordenados é geografica (como por exemplo latitude e longitude). Para calcular a área total do raster com um sistema de coordenadas projetado, encontramos o número total de pixels e multiplicamos pela área de cada pixel.

```
ramostra <- rast(eprdados::amostra_mapbiomas_2020)

total_pixels <- ncol(ramostra) * nrow(ramostra)
area_pixel_m2 <- res(ramostra)[1] * res(ramostra)[2]
area_paisagem_m2 <- total_pixels * area_pixel_m2
area_paisagem_m2
```

```
[1] 64337.77
```

E agora é so fazer a conversão de metros quadrados para hectares e quilômetros quadrados:

```
# hectares  
area_paisagem_m2 / 10000
```

```
[1] 6.433777
```

```
# quilometros quadrados  
area_paisagem_m2 / 1000000
```

```
[1] 0.06433777
```

1.2 Pergunta 2

Utilizando as funções disponíveis no pacote `tmap`, crie mapas temáticos dos objetos `ramostra_media` e `ramostra_modal`. Inclua cópias do seu código e mapas na sua resposta. Você pode usar o `printscreen` para mostrar o RStudio com seu código e mapas.

1.2.1 resposta

Para responder a pergunta você precisa trocar o nome dos objetos usado no código no capítulo. Por exemplo, segue o novo código para criar um mapa temático com o objeto `ramostra_media`. Neste caso a agregação com a média cria novos valores (incorretos), que não fazem parte da classificação MapBiomias. Assim sendo, também removemos as linhas de código que especificavam uma legenda com as classes MapBiomias.

```
tm_shape(ramostra_media) +  
  tm_raster(style = "cat") +  
  tm_grid(labels.format = list(big.mark = "")) +  
tm_compass(position = c("right", "bottom")) +  
tm_scale_bar(breaks = c(0, 0.05, 0.1), text.size = 1,  
             text.color = "white", position=c("right", "bottom")) +  
tm_layout(legend.position = c("right","top"),legend.bg.color = "white")
```

1.3 Pergunta 3

Confira o código e os resultados obtidos anteriormente, quando mudamos a resolução da raster amostra (por exemplo figura 1.4). Explique o que aconteceu. Como e porque mudou os valores em cada caso (média e modal)?

1.3.1 resposta

Para responder a pergunta você deve revisar o conteúdo no capítulo Escala, especificamente na seção Alterando a resolução.

Preciso esclarecer e apresentar a sua entendimento sobre porque valores incorretos foram produzidos e porque a proporção de floresta mudou.

1.4 Pergunta 4

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

1.4.1 resposta

Para responder a pergunta você deve revisar o conteúdo no capítulo Escala, especificamente:

1. A definição de escala e extensão no contexto de Ecologia da Paisagem.
2. A figura mostrando os “componentes de uma raster e suas atributos no pacote terra”.
3. Os exemplos de código mostrando “como obter informações específicas através de funções específicas”.

Com base nas informações que já conhecemos, existem diferentes opções para saber a extensão em número de pixels desse recorte `buffer.forest1.1km`. Como o raio da buffer usado para fazer o corte foi de 1 km podemos estimar sem R, assim:

- i) A extensão seria o diâmetro da buffer (raio $\times 2 = 2\text{km}$).
- ii) Sabemos que a resolução (comprimento) de cada pixel é cerca de 29,9 metros. Portanto, a extensão em número de pixels seria aproximadamente: $2000 \text{ metros} / 29,9 \text{ metros} = 66,89 \text{ pixels}$.

No R podemos obter o valor exato através funções de `ncol()` e `nrow()`, respectivamente para extensões no sentido leste-oeste e norte-sul, assim:

```
ncol(buffer.forest1.1km)
```

```
[1] 68
```

```
nrow(buffer.forest1.1km)
```

```
[1] 68
```

1.5 Pergunta 5

Usando os valores listados 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.

1.5.1 resposta

Com base nas informações que já conhecemos, existem diferentes opções para saber a proporção de floresta em cada uma das diferentes extensões de buffer. A proporção de floresta seria a área de floresta dividido por área total da buffer. Podemos estimar sem R, aqui ilustrando o processo com a buffer com raio de 500 metros:

1. A extensão seria o diâmetro da buffer (raio $\times 2 = 1000$ m).
2. Sabemos que a área de um círculo é π vezes o raio elevado ao quadrado. Portanto, estimar a área do buffer em metros quadrados: $\pi \times (\text{raio}^2) = 3.14 \times 250000 = 785000$.
3. Unidades diferentes (raio = metros, área em hectares), portanto calcular área do buffer em hectares. $785000 / 10000 = 78,5$ hectares.
4. Agora que temos tudo nas mesmas unidades podemos calcular a proporção de floresta. A proporção de floresta seria a área de floresta dividido por área total da buffer $= 6,3 / 78,5 = 0.08$. Ou seja, cerca de 8 % é floresta.

E depois, repetindo o mesmo processo para os outros valores. Isso seria possível tanto com um calculador, quanto com ferramentas de planilha como [Excel](#) e [Calc](#).

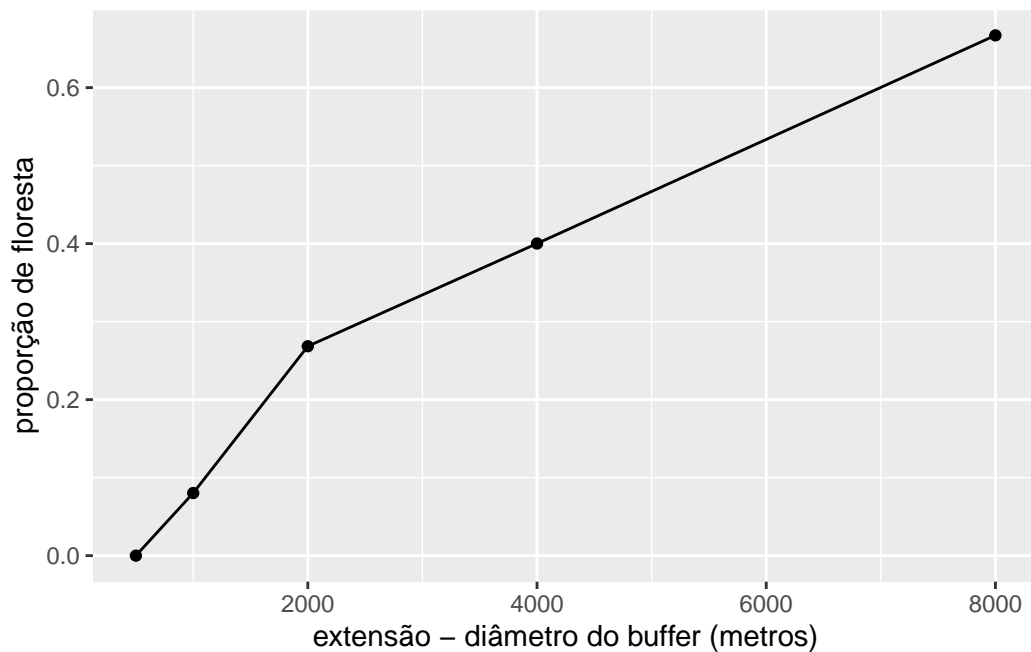
Podemos responder usando R em 2 passos assim:

```

# 1 construir uma data frame "raios" com os valores
raio_m <- c(250, 500, 1000, 2000, 4000)
floresta_ha <- c(0, 6.3, 84.3, 502.6, 3351)
raios <- data.frame(raio_m = raio_m,
                    floresta_ha = floresta_ha)

# 2 Calculos e grafico.
# Unidades diferentes (raio em metros, área em hectares).
# Portanto vamos calcular área de cada buffer em hectares
# e fazer o grafico
raios |>
  mutate(ext_m = raio_m*2,
         area_buff_m2 = (3.14 * raio_m^2)) |>
  mutate(area_buff_ha = area_buff_m2/10000) |>
  mutate(prop_floresta = floresta_ha/area_buff_ha) |>
  ggplot(aes(x = ext_m, y = prop_floresta)) +
  geom_point() +
  geom_line() +
  labs(x = "extensão - diâmetro do buffer (metros)",
       y = "proporção de floresta")

```



1.6 Pergunta 6

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.

1.6.1 resposta

2 Capítulo Métricas

Carregar os pacotes necessários:

```
library(tidyverse)
library(sf)
library(terra)
library(eprdados)
library(landscapemetrics)
```

2.1 Pergunta 1

Descreva brevemente 2 métricas de cada nível (patch, class, landscape) usando ajudar (usando ? e/ou list_lsm), aulas ("Métricas da paisagem: Modelo mancha-corredor-matriz" e "Índices de Paisagem e Análises de Padrões Espaciais") e/ou a leitura disponível no Google Classroom (Base teórica 4 Dados, métricas, análises). Incluindo na descrição - o nome, porque serve, unidades de medida, e relevância ecológica.

2.1.1 resposta

Para responder a pergunta você deve revisar as aulas e o conteúdo no capítulo Métrica. Métricas (descrição - o nome, porque serve, unidades de medida, e relevância ecológica) para todos os níveis foram apresentadas nas aulas "Métricas da paisagem: Modelo mancha-corredor-matriz" e "Índices de Paisagem e Análises de Padrões Espaciais". Exemplo com uma métrica tamanho (área):

1. Nível de mancha - tamanho da mancha.
2. Nível de classe - tamanho média das manchas.
3. Nível de paisagem - tamanho média das manchas.

Podemos verificar as métricas disponíveis no pacote [landscapemetrics](#) com o seguinte código:


```

# ajudar para entender a função
?list_lsm()
# Nível de mancha
landscapemetrics::list_lsm(level = "patch", type = "area and edge metric")
?landscapemetrics::list_lsm
# Nível de classe
landscapemetrics::list_lsm(level = "class", type = "area and edge metric")
?landscapemetrics::list_lsm
# Nível de paisagem
landscapemetrics::list_lsm(level = "landscape", type = "area and edge metric")
?landscapemetrics::list_lsm

# todos as métricas com area
landscapemetrics::list_lsm(metric = "area")

```

2.2 Pergunta 2

O modelo mancha-corredor-matriz é frequentemente adotado na ecologia da paisagem. Com base nas aulas teóricas e usando os valores no objeto `minha_amostra_1000` apresentados na tabela acima, identificar qual classe representar a matriz na paisagem. Há alguma informação faltando que limita a sua capacidade de identificar qual classe representar a matriz? Se sim, o que precisa ser adicionado? Justifique as suas respostas de forma clara e concisa.

2.2.1 resposta

Para responder a pergunta você precisa entender a definição da “matriz” no contexto de Ecologia da Paisagem - modelo de mancha-corredor-matriz (aula - Métricas da paisagem: Modelo mancha-corredor-matriz). Neste caso, class 0 é o classe dominante - com a área core cobrindo quase 67% da paisagem. Assim sendo, classe 0 é a matriz. Não há necessidade de informações adicionais. Se os valores estivessem menores e mais próximos (por exemplo, 41 e 44%), provavelmente precisaríamos incluir métricas adicionais para representar padrões espaciais na paisagem, como coesão, tamanho do maior mancha etca.

2.3 Pergunta 3

Em vez de extensão, você precisa incluir o tamanho (área do círculo) correspondente a cada raio. Incluir uma cópia do código ajustado para produzir uma figura com tamanho (área em quilômetros quadrados) no eixo x.

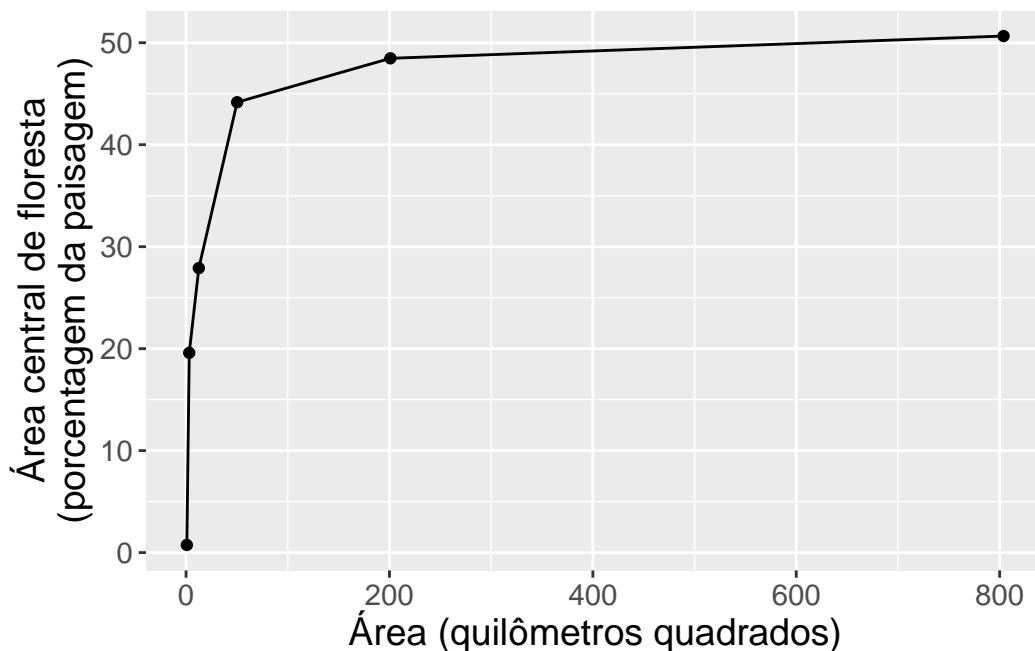
2.3.1 resposta

Seguindo o exemplo do capítulo, sabemos que os buffers têm formato de círculo, portanto:

1. A extensão seria o diâmetro da buffer (raio x 2).
2. Sabemos que a área de um círculo é pi vezes o raio elevado ao quadrado. Portanto, estimar a área do buffer em metros quadrados: $\pi \times (\text{raio}^2)$.
3. E para finalizar converta o valor da área em quilômetros quadrados.

No R, podemos fazer isso, ajustando o código no capítulo com a função `mutate()`. Usamos `mutate()` para fazer novas colunas com o valor da área em metros quadrados `area_m2` e quilômetros quadrados `area_km2`. Mantemos os nomes curtos, e depois atualize o rótulo do eixo para refletir a nova coluna com valores de área, especificando o nome do eixo `x` com a função `labs()`.

```
amostras_metrica |>
  filter(class==1) |>
  mutate(ext_m = 2*raio,
         area_m2 = 3.14*(raio^2)) |>
  mutate(area_km2 = area_m2/1000000) |>
  ggplot(aes(x = area_km2, y = value)) +
  geom_point() +
  geom_line() +
  labs(x = "Área (quilômetros quadrados)",
       y = "Área central de floresta\n(porcentagem da paisagem)") +
  theme(text = element_text(size = 14))
```



2.4 Pergunta 4

Em menos de 200 palavras apresente a sua interpretação do gráfico em figura 5.2.

2.4.1 resposta

A proporção de floresta muda com a escala espacial, representado pela extensão do buffer em torno do ponto amostral. Existe uma relação positiva entre o percentual de área central e a extensão. A proporção de floresta aumenta de zero a 51 por cento, mas a relação não é linear. Os valores aumentam rapidamente até extensões de 8 km (6% por quilometro em média entre 0 e 8 quilômetros). Após isso a curva parece atingir uma assíntota (aumentando 0.25% por quilometro entre 8 e 16 quilômetros).

2.5 Pergunta 5

Comparar os resultados apresentados nas figuras com modelos lineares e não-lineares. Como podemos estabelecer qual seria o melhor modelo? Qual modelo seria mais adequado para identificar limiares no padrão de área central de floresta?

2.5.1 resposta

Uma inspeção visual sugere que um modelo linear sem qualquer transformação tem os maiores intervalos de confiança e é o pior modelo. Um modelo linear com dados transformados parece ajustar-se próximo aos dados. O modelo não linear mostra melhor os padrões nos dados originais. Mas essas avaliações são subjetivas. Deveríamos realizar testes estatísticos para confirmar estas avaliações visuais. Todos os modelos melhorariam (por exemplo os intervalos de confiança ficariam menores) se aumentássemos o tamanho da amostra calculando a métrica com mais distâncias de buffer.

2.6 Pergunta 6

Aumentar o tamanho amostral na análise. Usando os exemplos de código anteriores, aumentar o número de buffers entre 125 m e 16 km. i) Calcular a métrica cpland para 12 distâncias de buffers e montar dados novas (isso é cinco distâncias novas e as sete distâncias originais) em um objeto com nome de "amostras_metrica_nova". ii) usando ggplot2 fazer um gráfico com os novos dados mostrando valores de extensão no eixo x e proporção da floresta central no eixo y. Ajustar o código para incluir intervalos de confiança junto com o modelo linear. Comparar os resultados apresentados nas figuras com modelos lineares sem transformação. O que aconteceu com o ajuste do modelo e os intervalos de confiança? Foi útil incluir mais distâncias? Justifique as suas respostas de forma clara e concisa. Inclua cópias do seu código e gráficos na sua resposta. Você pode usar o printscreen para mostrar o RStudio com seu código e gráficos.

2.6.1 resposta

Passos para desenvolver uma resposta:

1. Decida as distâncias para os seis novos buffers.
2. Calcular a métrica nos novos buffers.
3. Adicione os novos dados junto com as seis distâncias originais.
4. Faça o gráfico.

Olhando o gráfico "Comparação de padrões lineares e não-lineares." vemos que quando os pontos estão mais próximos os intervalos de confiança são menores. Os intervalos de confiança são maiores onde há maiores distâncias entre os pontos. Portanto, incluiremos mais buffers entre 1 e 16 quilômetros.

O código abaixo fará isso em R:

Lembrando o código para as buffers originais:

```
# raio 250 metros
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 250, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 250) -> minha_amostra_250
# raio 500 metros
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 500, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 500) -> minha_amostra_500
# raio 1 km (1000 metros)
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 1000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 1000) -> minha_amostra_1000
# raio 2 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 2000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 2000) -> minha_amostra_2000
# raio 4 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 4000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 4000) -> minha_amostra_4000
# raio 8 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 8000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 8000) -> minha_amostra_8000
# raio 16 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 16000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 16000) -> minha_amostra_16000
```

Podemos agora seguir o exemplo original e incluir novas distâncias (1,5km; 3km; 6km; 11km e 14km):

```

# novas buffers
# raio 1.5 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 1500, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 1500) -> minha_amostra_1500
# raio 3 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 3000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 3000) -> minha_amostra_3000
# raio 6 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 6000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 6000) -> minha_amostra_6000
# raio 11 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 11000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 11000) -> minha_amostra_11000
# raio 14 km
sample_lsm(floresta_2020, y = rio_pontos_31976[1, ],
           size = 14000, shape = "circle",
           metric = "cpland") |>
  mutate(raio = 14000) -> minha_amostra_14000

# juntar dados objeto novo com nome de "amostras_metrica_nova"
bind_rows(minha_amostra_250,
          minha_amostra_500,
          minha_amostra_1000,
          minha_amostra_1500,
          minha_amostra_2000,
          minha_amostra_3000,
          minha_amostra_4000,
          minha_amostra_6000,
          minha_amostra_8000,
          minha_amostra_11000,
          minha_amostra_14000,
          minha_amostra_16000) -> amostras_metrica_nova

```

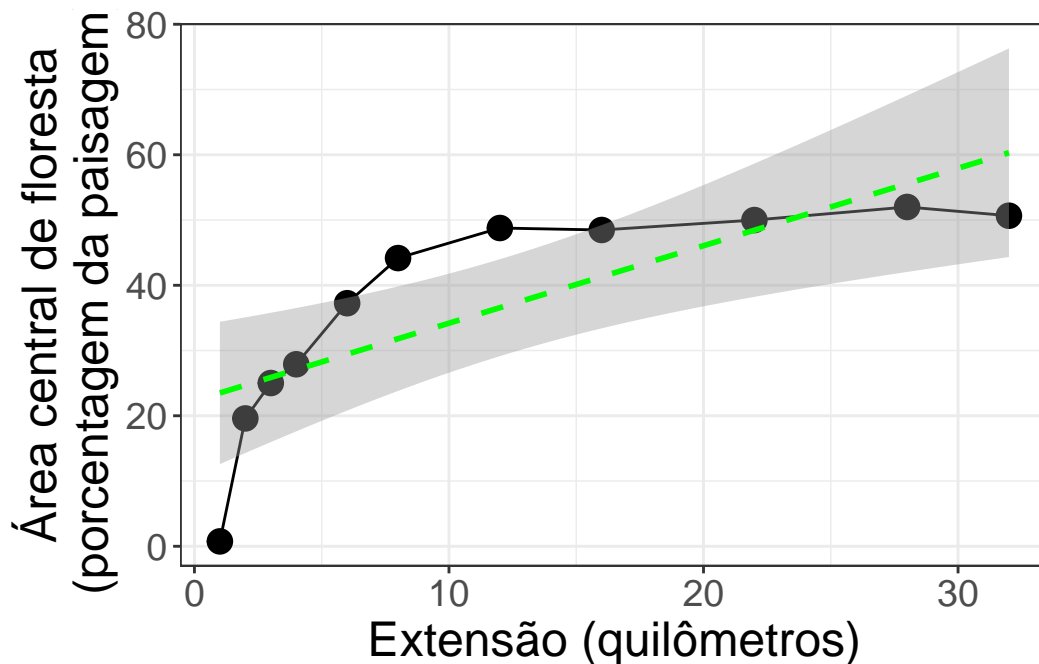
Grafico.

```

amostras_metrica_nova |>
  filter(class==1) |>
  mutate(ext_m = 2*raio,
         ext_km = (2*raio)/1000) |>
# fazer o grafico
ggplot(aes(x = ext_km, y = value)) +
  geom_point(size = 4) +
  geom_line() +
  stat_smooth(method = "lm", se = TRUE, color = "green",
             linetype = "dashed") +
  labs(x = "Extensão (quilômetros)",
       y = "Área central de floresta\n(porcentagem da paisagem)") +
  theme_bw() +
  theme(text = element_text(size = 18))

```

`geom_smooth()` using formula = 'y ~ x'



2.7 Pergunta 7

Com base nos resultados apresentados (figura e tabela) caracterizar as mudanças na paisagem em função de extensões diferentes. Olhando os graficos prever como

seria o padrão para extensões maiores (lembrando que valores são dobrados - por exemplo raio de 250 metros gerar uma extensão de 500 metros). Seria relevante repetir incluindo calculos para extensões maiores (por exemplo 64 km e 128 km)? Justifique sua caracterização e previsões de forma clara e concisa, apoie sua escolha com exemplos da literatura científica.

2.7.1 resposta

2.8 Pergunta 8

Usando como base o conteúdo das aulas, leitura disponível no Google Classroom (Base teórica 4 Dados, métricas, análises), e/ou exemplos apresentados aqui no tutorial, selecione pelo menos oito métricas de nível classe para caracterizar a paisagem de estudo e objetivos da sua projeto em grupo. Justifique sua seleção de forma clara e concisa, apoie sua escolha com exemplos da literatura científica.

2.8.1 resposta

References