

# Métricas da paisagem

Darren Norris

2022-09-29

## Sumário

<b>1</b>	<b>Apresentação</b>	<b>2</b>
<b>2</b>	<b>Código e R</b>	<b>2</b>
2.1	Organização do código no tutorial . . . . .	3
2.2	Métricas da paisagem e pacote “landscapemetrics” . . . . .	3
2.2.1	Pergunta 1 . . . . .	4
2.3	Pacotes . . . . .	4
<b>3</b>	<b>Dados</b>	<b>4</b>
<b>4</b>	<b>Cálculo de métricas</b>	<b>8</b>
4.1	Ponto único, raio único, métrica única . . . . .	8
4.1.1	Pergunta 2 . . . . .	8
4.2	Ponto único, distâncias variadas, métrica única . . . . .	9
4.2.1	Faça um gráfico . . . . .	12
4.2.2	Pergunta 3 . . . . .	13
4.2.3	Faça um gráfico elegante . . . . .	13
4.2.4	Pergunta 4 . . . . .	13
4.2.5	Compare linear and non linear . . . . .	14
4.2.6	Pergunta 5 . . . . .	14
4.3	Ponto único, distâncias variadas, métricas variadas . . . . .	15
4.3.1	Pergunta . . . . .	17

# 1 Apresentação

Nesta aula (...) vamos .... na ecologia da paisagem através cálculos com a proporção de floresta. Durante a aula você aprenderá a

## 2 Código e R

- Objetivo não é de apresentar detalhes sobre os cálculos/métodos estatísticas ou as funções no [R](#). Existem diversos exemplos disponíveis “[Ciência de Dados com R–Introdução.....](#)”: e com google “r cran introdução tutorial”..... Além disso, existem grupos de ajuda, como por exemplo: [R Brasil](#) e [Stack Overflow em Português](#)
- O objetivo é de apresentar um tutorial mostrando as capacidades e opções para desenvolver e integrar pesquisas na ecologia da paisagem no ambiente estatístico de [R](#)

Porque use R? R tem a capacidade (baseada em código) para alternar entre tarefas de processamento, modelagem e visualização de dados geográficos e não geográficos. Além disso, como é possível importar, modificar, analisar e visualizar dados espaciais no mesmo ambiente com script/código, 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](#) inclui análises usando R.

## 2.1 Organização do código no tutorial

O tutorial 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
```

Além disso, os símbolos `#>` e/ou `[1]` no início de uma linha indica o resultado que você verá no console de R depois de rodar o código, como no próximo exemplo.

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

```
[1] 2
```

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

```
[1] 2
```

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

```
[1] 3
```

## 2.2 Métricas da paisagem e pacote “landscapemetrics”

As métricas de paisagem são a forma que os ecólogos de paisagem usam para descrever os padrões espaciais de paisagens para depois avaliar a influência destes padrões espaciais nos padrões e processos ecológicos.

landscapemetrics é um pacote R para calcular métricas de paisagem em paisagens categóricas (onde tem uma classificação de cobertura de terra/habitat), em um fluxo de trabalho organizado. O pacote pode ser usado como um substituto do FRAGSTATS (McGarigal et al. 1995 <https://doi.org/10.2737/PNW-GTR-351>), pois oferece um fluxo de trabalho reproduzível para análise de paisagem em um único ambiente. Também permite cálculos de quatro métricas teóricas de complexidade da paisagem: entropia marginal, entropia condicional, entropia conjunta e informação mútua (Nowosad e Stepinski 2019 <https://doi.org/10.1007/s10980-019-00830-x>).

Nesse pacote o formato geral para uma função é o seguinte: A primeira parte é sempre `lsm_` (“landscape-metric”), seguida do “nível\_” e por fim a “métrica”. Ou seja, todas as funções começam com `lsm_`, daí você deve incluir o nível da análise “p” para patch (ou seja, para a mancha ou fragmento), “c” para classe e “l” para landscape ou seja, para métricas para a paisagem como um todo. E daí existem inúmeras métricas, como por exemplo a `cpland` (percentual de área central - “core area”) na paisagem, como vimos na aula teórica. Digite o código abaixo e veja o resultado. Leia com atenção e preste particular atenção na organização da página de ajuda.

```
library(landscapemetrics)
?landscapemetrics
```

No final da página você vai encontrar a palavra “Index”. Clique nela e você verá todas as funções do pacote. Desça até as `lsm_`. . . e clique em algumas delas ali. Explorar!

### 2.2.1 Pergunta 1

Descreva brevemente 2 métricas de cada nível.

---

## 2.3 Pacotes

Além do “landscapemetrics”, precisamos carregar alguns pacotes a mais para facilitar a organização e apresentação de dados espaciais (vector e raster) e os resultados.

Carregar pacotes (que deve estar instalado antes):

```
library(tidyverse)
library(sf)
library(terra)
library(tmap)
library(gridExtra)
library(kableExtra)
library(mgcv)
```

## 3 Dados

Precisamos o arquivo com os dados de MapBiomias “`amostra_mapbiomas_2020.tif`”, que vocês baixaram no tutorial anterior (Escala <https://rpubs.com/darren75/escala>). Nós podemos carregar os dados de cobertura da terra “`amostra_mapbiomas_2020.tif`” com a função `rast`.

```
# Selecionar e carregar arquivo "amostra_mapbiomas_2020.tif"
ramostra <- rast(file.choose())
# 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
```

Plotar para verificar, incluindo nomes e as cores para classes de floresta (valor = 1) e não-floresta (valor = 0).

```
# 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_layout(legend.bg.color = "white")
```

Se esta todo certo, voces devem ter uma imagem parecida como o seguinte:

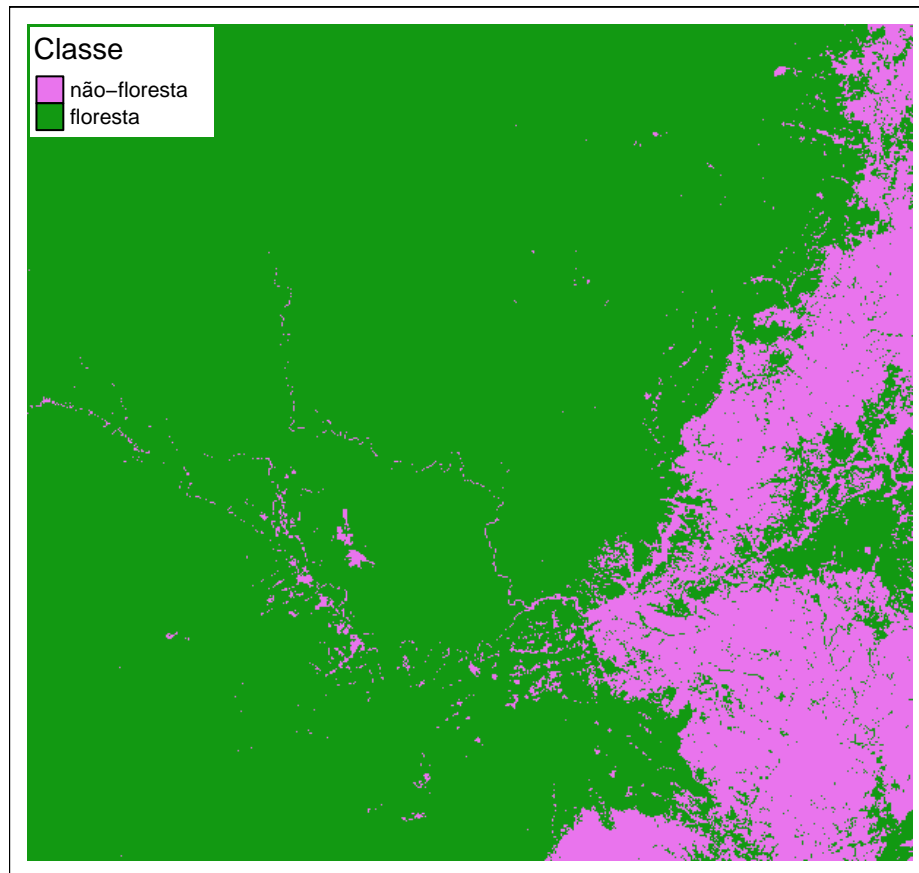


Figura 3.1: Floresta ao redor do Rio Araguari. MapBiomass 2020 reclassificado em floresta e não-floresta.

Agora temos a paisagem, precisamos também os pontos de amostra. Por isso, precisamos carregar os dados de rios e pontos de amostragem que usamos no tutorial Escala - arquivo "rivers.GPKG". Vamos carregar as camadas que vocês baixaram no tutorial anterior. Seleccionando o arquivo "rivers.GPKG", e carregando as camadas "midpoints" e "centerline".

```
# Selecionar o arquivo "rivers.GPKG",
meuSIG <- file.choose()
# Carregar pontos cada 5 km, camada midpoints
rsm_31976 <- sf::st_read(meuSIG, layer = "midpoints") %>%
  st_transform(31976)
# Carregar linha central de rios, camada centerline
rsl_31976 <- sf::st_read(meuSIG, layer = "centerline") %>%
  st_transform(31976)
```

Visualizar 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_31976) +
  tm_lines(col="blue") +
tm_shape(rsm_31976) +
  tm_dots(size = 0.2, col = "yellow") +
tm_layout(legend.bg.color="white")
```

Depois de executar (“run”) o código acima, você deverá ver a figura a seguir.

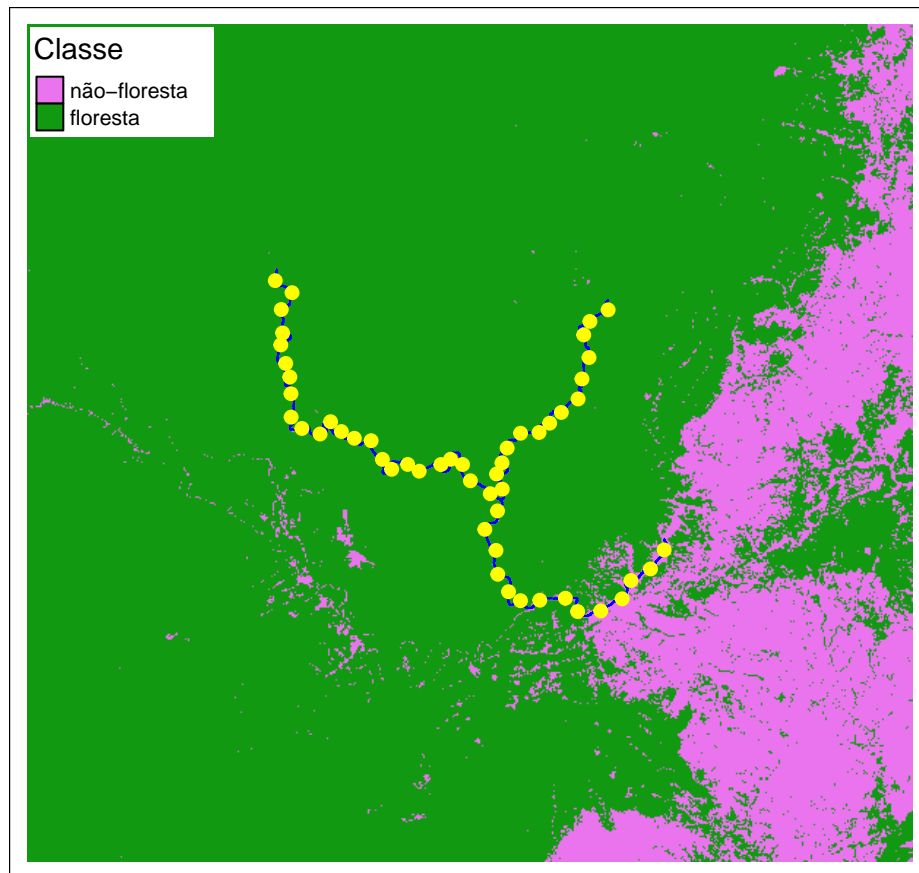


Figura 3.2: Cobertura da terra ao redor do Rio Araguari em 2020. Mostrando os pontos de amostragem (pontos amarelas) cada 5 quilômetros ao longo do rio (linha azul).

## 4 Cálculo de métricas

Para ilustrar como rodar as funções e cálculos com `landscapemetrics`, vamos calcular a área central na paisagem que usamos no tutorial de escala. Vamos estudar uma classe (floresta), portanto vamos incluir as métricas para nível de classe. Além disso, as métricas de paisagem em nível de classe são mais eficazes na definição de processos ecológicos (Tischendorf, L. Can landscape indices predict ecological processes consistently?. *Landscape Ecology* 16, 235–254 (2001). <https://doi.org/10.1023/A:1011112719782>).

Para calcular as métricas de paisagem dentro de um certo buffer em torno de pontos de amostra, existe a função `sample_lsm()`. Através da função `sample_lsm()` podemos calcular mais de 50 métricas da paisagem, dentro de extensões diferentes.

Para a função `sample_lsm()` funcionar, precisamos informar (i) a paisagem (arquivo de raster), (ii) ponto (vector), (iii) raio, (iv) forma do buffer (círculo ou quadrado) e por final (v) a métrica desejada.

### 4.1 Ponto único, raio único, métrica única

Métricas de área central (“core area”) são consideradas medidas da qualidade de habitat, uma vez que indica quanto existe realmente de área efetiva de um fragmento, após descontar-se o efeito de borda. Vamos calcular a percentual de área central (“core area”) no entorno de um ponto de amostragem. Isso seria, a percentual de áreas centrais (excluídas as bordas de 30 m) de cada classe em relação à área total da paisagem.

```
minha_amostra_1000 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 1000, shape = "circle",
                                metric = "cpland",
                                edge_depth = 1)
```

Depois que executar (“run”), podemos olhar os dados com o código a seguir.

```
minha_amostra_1000
```

Os dados deve ter os valores:

layer	level	class	id	metric	value	plot_id	percentage_inside
1	class	0	NA	cpland	66.94191	1	99.9608
1	class	1	NA	cpland	19.07745	1	99.9608

#### 4.1.1 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.



## 4.2 Ponto único, distâncias variados, métrica única

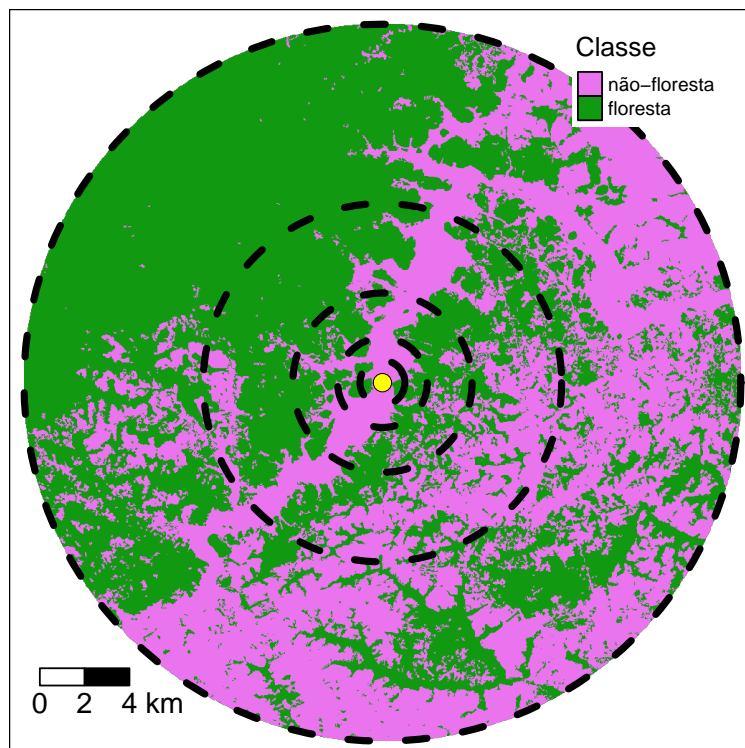


Figura 4.1: Cobertura florestal em extensões diferentes ao redor de um ponto de amostragem.

Para uma comparação multiescala, vamos calcular a mesma métrica, no mesmo ponto, mas agora com extensões diferentes. Continuando o exemplo no tutorial anterior (Escala), vamos repetir o mesmo processo, mas agora com raios de 250, 500, 1000, 2000, 4000, 8000 e 16000 metros, dobrando a escala (extensão) em cada passo.

Para obter resultados com extensões diferentes, precisamos primeiramente repetir o código, ajustando para cada extensão, e depois juntar os resultados. O código a seguir calculará a mesma métrica para as diferentes distâncias.

```
# raio 250 metros
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 250, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 250) -> minha_amostra_250
# raio 500 metros
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 500, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 500) -> minha_amostra_500
# raio 1 km (1000 metros)
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 1000, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 1000) -> minha_amostra_1000
# raio 2 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 2000, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 2000) -> minha_amostra_2000
# raio 4 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 4000, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 4000) -> minha_amostra_4000
# raio 8 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 8000, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 8000) -> minha_amostra_8000
# raio 16 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 16000, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 16000) -> minha_amostra_16000
```

E agora, o código a seguir juntará os resultados das diferentes extensões.

```
bind_rows(minha_amostra_250,  
          minha_amostra_500,  
          minha_amostra_1000,  
          minha_amostra_2000,  
          minha_amostra_4000,  
          minha_amostra_8000,  
          minha_amostra_16000) -> amostras_metrica
```

Depois que executar (“run”), podemos olhar os dados “amostras\_metrica” com o código a seguir.

```
amostras_metrica
```

Os dados deve ter os valores (coluna value) da métrica (coluna metric) de cada classe (coluna class) para cada distância (coluna raio):

layer	level	class	id	metric	value	plot_id	percentage_inside	raio
1	class	0	NA	cpland	79.4	1	99	250
1	class	0	NA	cpland	86.9	1	100	500
1	class	1	NA	cpland	0.7	1	100	500
1	class	0	NA	cpland	66.9	1	100	1000
1	class	1	NA	cpland	19.1	1	100	1000
1	class	0	NA	cpland	57.6	1	100	2000
1	class	1	NA	cpland	26.9	1	100	2000
1	class	0	NA	cpland	36.2	1	100	4000
1	class	1	NA	cpland	42.2	1	100	4000
1	class	0	NA	cpland	35.8	1	100	8000
1	class	1	NA	cpland	45.4	1	100	8000
1	class	0	NA	cpland	37.1	1	100	16000
1	class	1	NA	cpland	46.3	1	100	16000

### 4.2.1 Faça um gráfico

Uma imagem vale mais que mil palavras. Portanto, gráficos/figuras/imagens são uma das mais importantes formas de comunicar a ciência. Os dados apresentados em uma tabela podem ser difíceis de entender. Portanto, a primeira pergunta que você deve se fazer é se você pode transformar aquela tabela (chata e feia) em algum tipo de gráfico. Lembrando, sempre pode incluir a tabela como anexo.

Aqui, vamos fazer um gráfico com os dados `amostras_metrica`, usando o pacote `ggplot2`.

O `ggplot2` faz parte do conjunto “tidyverse”, e é um pacote de visualização de dados. “gg” se refere a uma gramática de gráficos. A ideia principal é criar um gráfico como se fosse uma frase de uma língua, onde cada elemento do gráfico seria uma palavra, organizados em uma sequência lógica para construir uma frase completa (gráfico final). Isto nos permite construir gráficos tão complexos quanto quisermos. Os gráficos criados com `ggplot2` são, em geral, mais elegantes do que os gráficos tradicionais do R.

O `ggplot2` exige que os dados a serem plotados estejam em um data frame. Ou seja, sempre teremos que transformar os dados para data frame ou construir um data frame com os dados que possuímos.

O comando principal função a ser utilizado se chama `ggplot` (sem o “2”).

Para `ggplot`, precisamos os dados (data frame), e depois cria o “mapeamento” das variáveis, normalmente usando “aes” (de aesthetics). Ou seja, você especifica quais são as variáveis dos eixos x e y dentro de aes. Através dele vamos definir qual é a variável preditora/explanadora (eixo x) e qual é a variável resposta (eixo y) em nosso conjunto de dados.

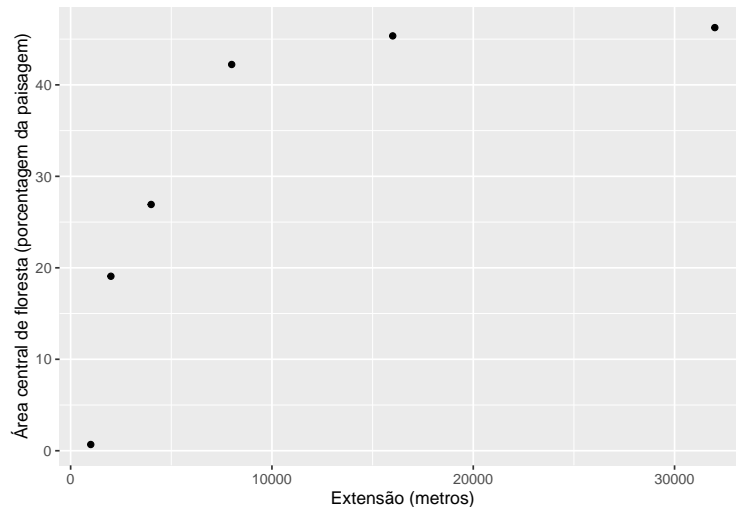
Depois da função `ggplot`, na sequência no código nós especificamos qual tipo de gráfico com um “geom”. Por exemplo, `geom_point` para plotar pontos, `geom_boxplot` para um boxplot, etc. Para a lista completa de geoms e todas as outras opções do pacote, visite a página do projeto `ggplot2`, livro e sempre exemplos no google, por exemplo digitando: `ggplot2` gráfico de barra no Google, tem mais de 50 mil resultados com páginas de imagens, código pronto e exemplos no YouTube.

Note que adicionamos um geom com um “+”. No `ggplot2`, nós criamos gráficos em camadas, e adicionamos camada a camada com um “+”. Assim, é possível ajustar qualquer elemento do gráfico.

Aqui vamos fazer um gráfico com valores de extensão no eixo x e proporção da floresta central no eixo y. Assim sendo, com o código a seguir, vamos informar (i) os dados (selecione classe de floresta através de um filtro), (ii) as colunas para os eixos x e y, (iii) tipo de gráfico (gráfico de pontos), (iv) nomes para os eixos.

```
amostras_metrica %>%
  filter(class==1) %>%
  ggplot(aes(x=2*raio, y=value)) +
  geom_point() +
  labs(x = "Extensão (metros)",
       y = "Área central de floresta (porcentagem da paisagem)")
```

Depois de executar (“run”) o código acima, você deverá ver o gráfico a seguir.



#### 4.2.2 Pergunta 3

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

---

#### 4.2.3 Faça um gráfico elegante

Podemos ajustar qualquer elemento do gráfico. Agora, vamos mudar as unidades de metros para quilômetros, aumentar o tamanho dos pontos, incluir uma linha reta para ilustrar a tendência geral, colocar o título longo do eixo y em duas linhas, e aumentar o tamanho da fonte para o texto ficar mais claro.

```
amostras_metrica %>%
  filter(class==1) %>%
  ggplot(aes(x=(2*raio)/1000, y=value)) +
  geom_point(size = 4) +
  stat_smooth(method = "lm", se = FALSE, color = "green") +
  labs(x = "Extensão (quilômetros)",
       y = "Área central de floresta\n(porcentagem da paisagem)") +
  theme(text = element_text(size = 18))
#> `geom_smooth()` using formula 'y ~ x'
```

#### 4.2.4 Pergunta 4

Em menos de 200 palavras apresente a sua interpretação do gráfico em figura 4.2.

---

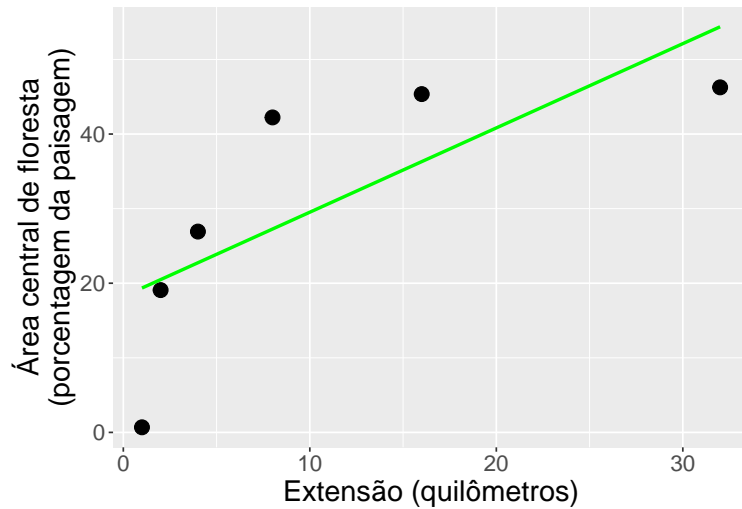


Figura 4.2: Comparação da área central de floresta em diferentes extensões.

#### 4.2.5 Compare linear and non linear

include figure

more text

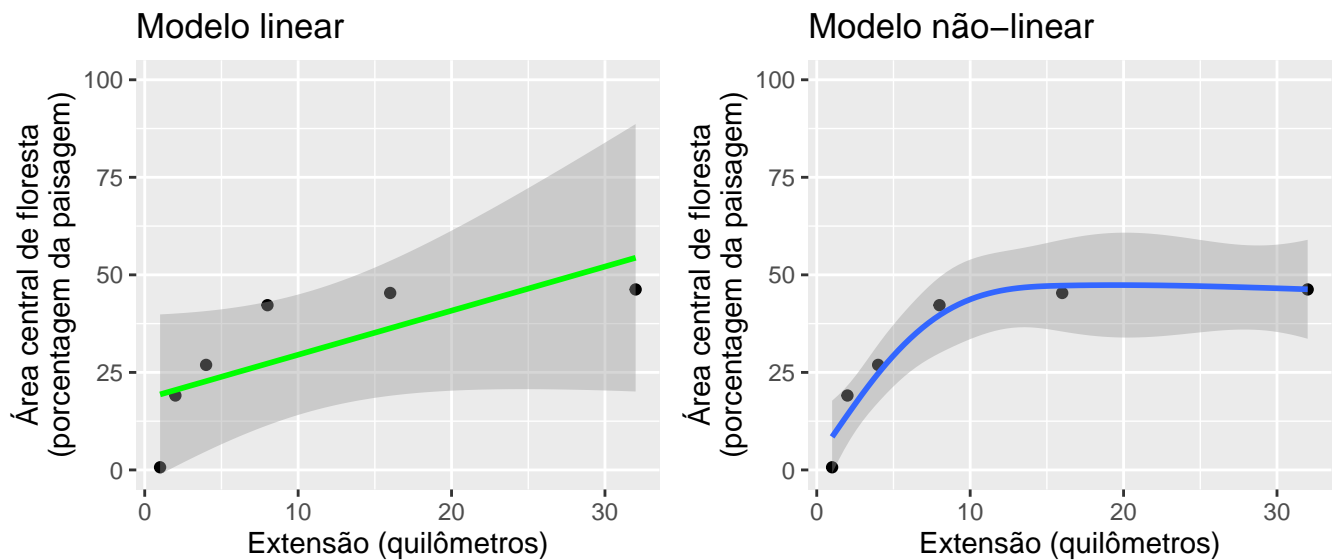


Figura 4.3: Comparação de padrões lineares e não-lineares.

#### 4.2.6 Pergunta 5

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

### 4.3 Ponto único, distâncias variadas, métricas variadas

Como as mudanças na estrutura da paisagem caracterizam-se por serem não-lineares, para desenvolver análises estatísticas e evidências robustas pode (i) aplicar uma transformação (por exemplo, “log”) ou (ii) adotar modelos não-lineares. Aqui, usaremos modelos não lineares para comparar padrões em diferentes métricas da paisagem em diferentes extensões.

Plot extensions include figure

more text

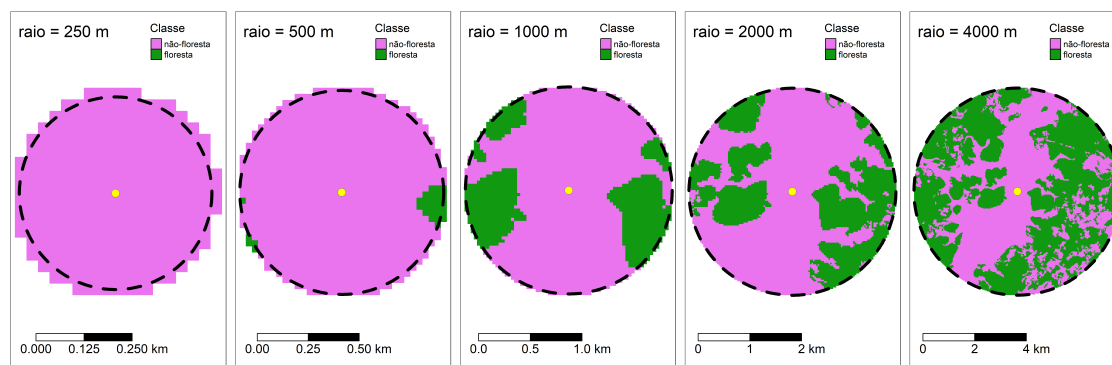


Figura 4.4: Cobertura florestal em extensões diferentes ao redor de um local de amostragem.

see what happens

---

Multiplwe metrcas area for all exten... sample\_lsm()

Aqui vamos .....

- Métricas de área e borda. Quantificam a composição da paisagem e fornecem sobre ela informações importantes sobre a dinâmica de populações vegetais e animais  
pland = area and edge metric / percentage of landscape  
percentage da paisagem  
Porcentagem de cobertura da classe na paisagem.  
ed = area and edge metric / edge density . densidade de borda que é igual à soma dos comprimentos (m) de todos os segmentos de borda que envolvem o fragmento, dividida pela área total da paisagem (m<sup>2</sup>), sendo posteriormente convertido em hectares.
- Métricas de área central. Quantificam a composição da paisagem. cpland = core area metric / core area percentage of landscape / (percentual de área central (“core”) na paisagem)  
Percentual de áreas centrais (excluídas as bordas de 30 m) em relação à área total da paisagem. O termo “Core area” foi traduzido como área central ou área núcleo. Aqui vamos adotar área central.
- Métricas de contágio e agregação. Quantificam a configuração da paisagem: pd = aggregation metric / patch density  
cohesion = aggregation metric / patch cohesion index.

```
minhas_metricas <- c("pland", "ed", "cpland", "pd", "cohesion")

# raio 250 metros
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 250, shape = "circle",
```

```

    metric = minhas_metricas) %>%
  mutate(raio = 250) -> metricas_amostra_250
# raio 500 metros
sample_lsm(floresta_2020, y = rsm_31976[1, ],
  size = 500, shape = "circle",
  metric = minhas_metricas) %>%
  mutate(raio = 500) -> metricas_amostra_500
# raio 1 km (1000 metros)
sample_lsm(floresta_2020, y = rsm_31976[1, ],
  size = 1000, shape = "circle",
  metric = minhas_metricas) %>%
  mutate(raio = 1000) -> metricas_amostra_1000
# raio 2 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
  size = 2000, shape = "circle",
  metric = minhas_metricas) %>%
  mutate(raio = 2000) -> metricas_amostra_2000
# raio 4 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
  size = 4000, shape = "circle",
  metric = minhas_metricas) %>%
  mutate(raio = 4000) -> metricas_amostra_4000
# raio 8 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
  size = 8000, shape = "circle",
  metric = minhas_metricas) %>%
  mutate(raio = 8000) -> metricas_amostra_8000
# raio 16 km
sample_lsm(floresta_2020, y = rsm_31976[1, ],
  size = 16000, shape = "circle",
  metric = minhas_metricas) %>%
  mutate(raio = 16000) -> metricas_amostra_16000

```

join

```

bind_rows(metricas_amostra_250,
  metricas_amostra_500,
  metricas_amostra_1000,
  metricas_amostra_2000,
  metricas_amostra_4000,
  metricas_amostra_8000,
  metricas_amostra_16000) -> amostras_metricas

```

get forest

```

amostras_metricas %>%
  filter(class==1) -> amostras_metricas_floresta

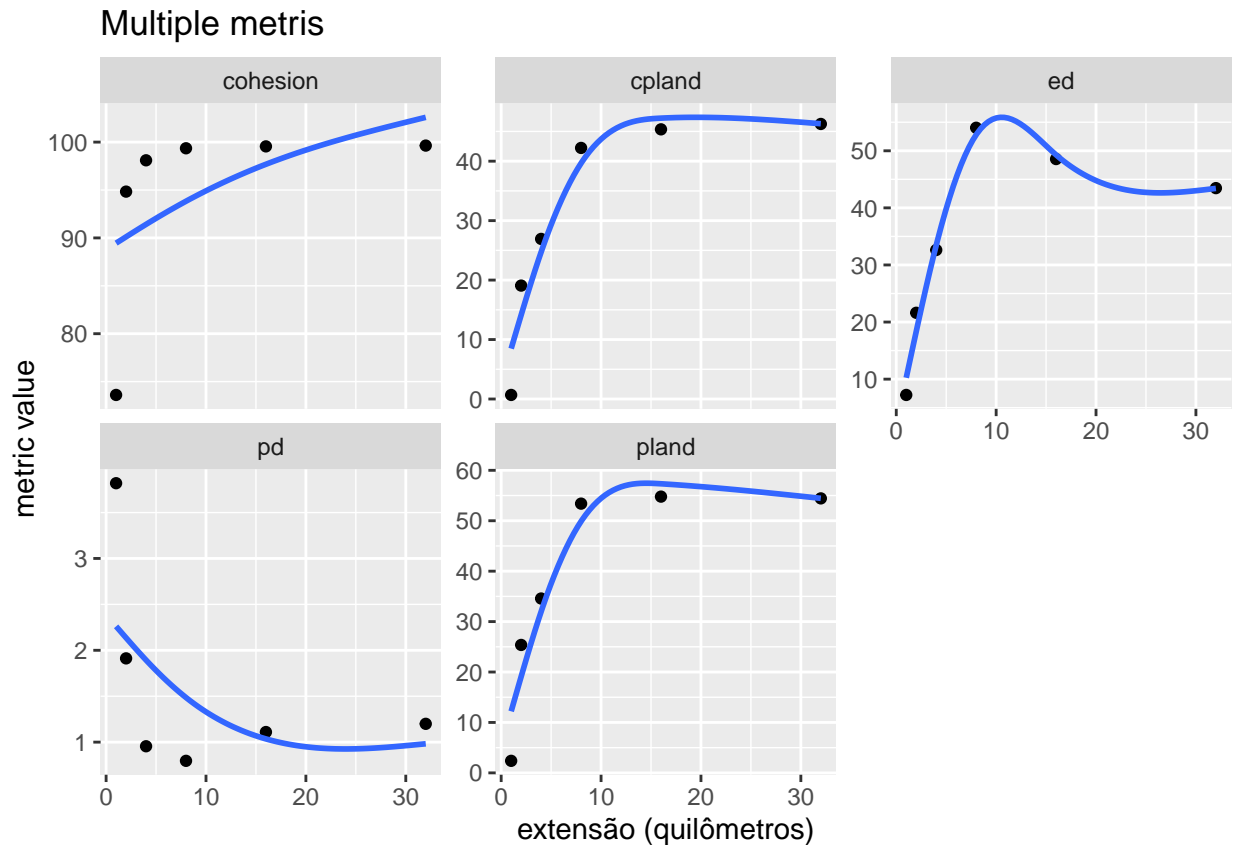
```

Best to include forest metrics 250 as 0 or NA?

Plot



```
amostras_metricas_floresta %>%
  ggplot(aes(x=(2*raio)/1000, y=value)) +
  geom_point() +
  stat_smooth(method = "gam", formula = y ~ s(x, k = 5),
             se=FALSE) +
  facet_wrap(~metric, scales = "free_y") +
  labs(title = "Multiple metris",
       x = "extensão (quilômetros)",
       y = "metric value")
```



#### 4.3.1 Pergunta

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 seis métricas de nível classe para caracterizar a paisagem de estudo e objetivos da sua projeto. Justifique sua seleção de forma clara e concisa, apoie sua escolha com exemplos da literatura científica.