

Métricas da paisagem

Sumário

1	Packages	1
2	Data	1
2.1	Plot	2
3	Area within buffer	3
3.1	Ponto único, raio único	3
3.2	Ponto único, distâncias variados	4
3.3	Ponto único, distâncias variados, métricas variadas	8

1 Packages

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

2 Data

load raster

River lines

Load

```
# pontos cada 5 km
rsm_31976 <- sf::st_read(meuSIG, layer = "midpoints") %>%
  st_transform(31976)
# linha central de rios
rsl_31976 <- sf::st_read(meuSIG, layer = "centerline") %>%
  st_transform(31976)
```

2.1 Plot

```
# 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_31976) +
  tm_lines(col="blue") +
tm_shape(rsm_31976) +
  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")
```

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

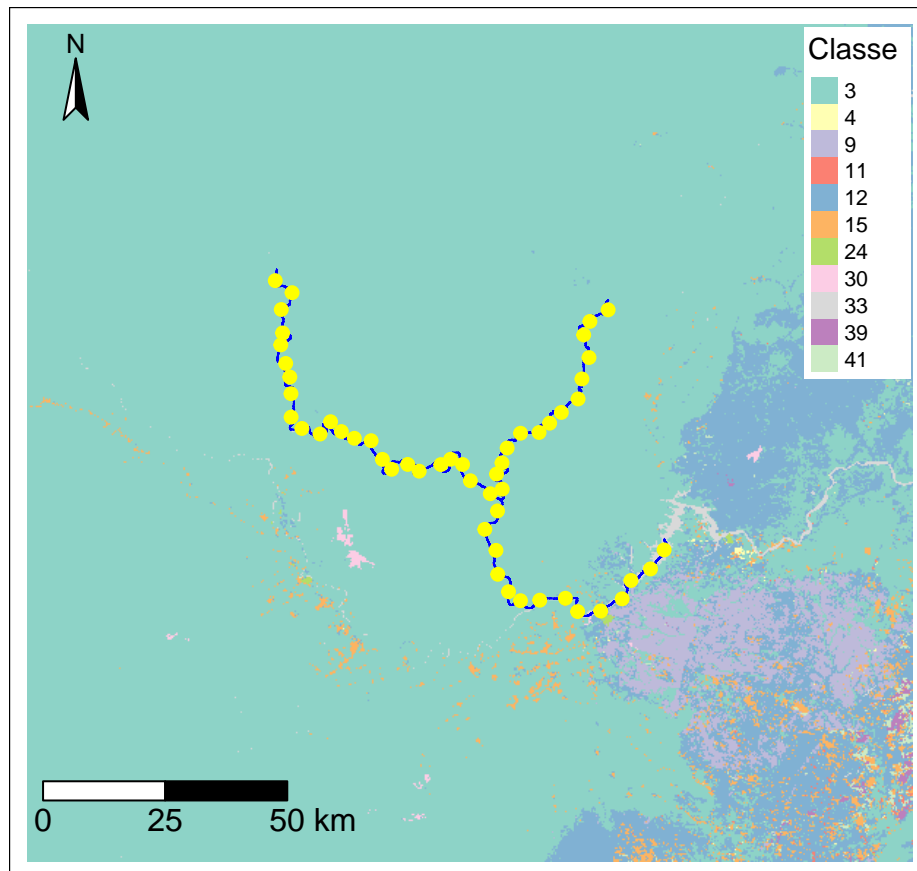


Figura 2.1: Cobertura da terra ao redor do Rio Araguari em 2020. Mostrando os pontos de amostragem (pontos amarelas) cada 5 quilômetros ao longo do rio.

3 Area within buffer

3.1 Ponto único, raio único

Para amostrar 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.

`cpland` Core area percentage of landscape (Core area metric) C%LAND (percentual de área central (“core”) na paisagem) Percentual de áreas centrais (excluídas as bordas de 30 m) em relação à área total da paisagem. Métricas de área central (“core”) É considerada medida da qualidade de hábitat, uma vez que indica quanto existe realmente de área efetiva de um fragmento, após descontar-se o efeito de borda.

```
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

3.2 Ponto único, distâncias variados

Core area for all exten...

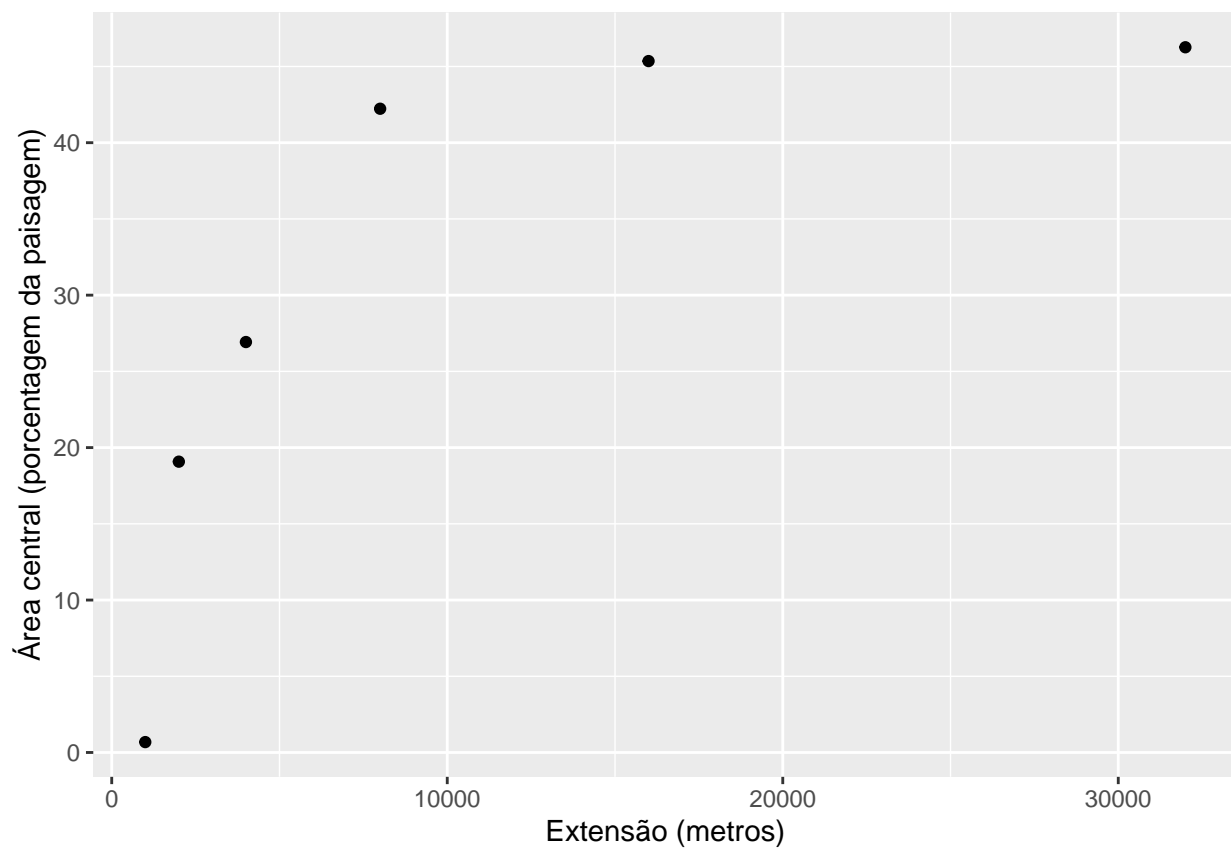
```
# raio 250 metros
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 250, shape = "circle",
           metric = "cpland") %>%
  mutate(raio = 250) -> minha_amostra_250
# raio 500 metros
sample_lsm(floresta_2020, y = rsm_31976[1, ],
           size = 500, shape = "circle",
           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
```

join

```
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
```

3.2.1 Make a graph

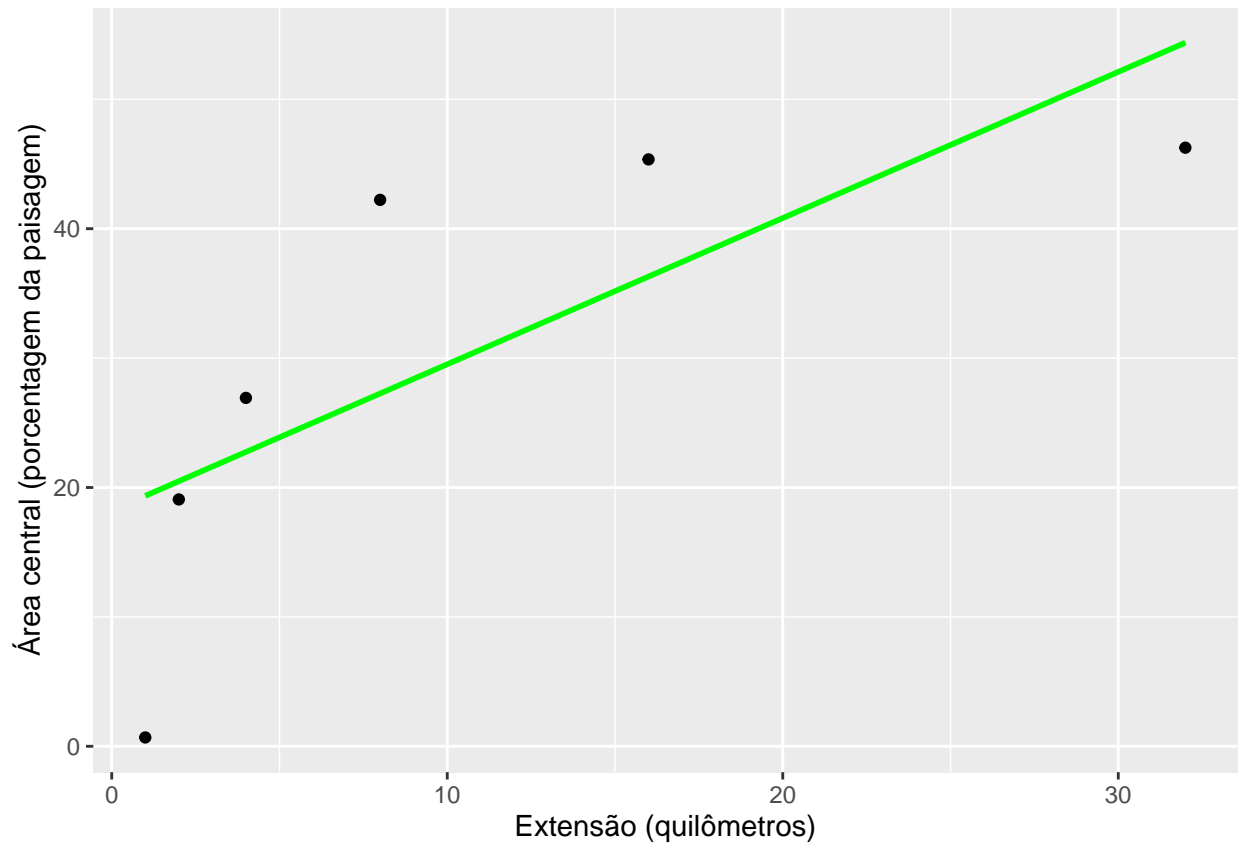
```
amostras_metrica %>%  
  filter(class==1) %>%  
  ggplot(aes(x=2*raio, y=value)) +  
  geom_point() +  
  labs(x = "Extensão (metros)", y = "Área central (porcentagem da paisagem)")
```



3.2.2 Make a better graph

```
amostras_metrica %>%  
  filter(class==1) %>%
```

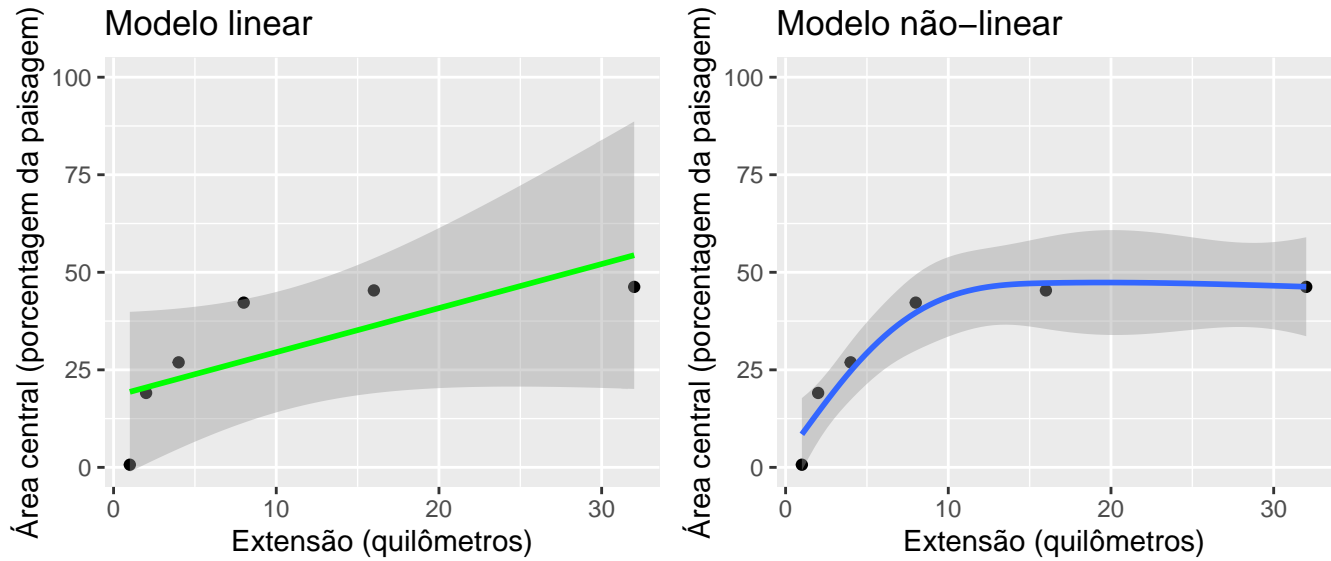
```
ggplot(aes(x=(2*raio)/1000, y=value)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE, color = "green") +
  labs(x = "Extensão (quilômetros)", y = "Área central (porcentagem da paisagem)")
#> `geom_smooth()` using formula 'y ~ x'
```



3.2.3 Compare linear and non linear

include figure

more text



see what happens

3.3 Ponto único, distâncias variados, métricas variadas

Multiplwe metricas 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 percentagem 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 (m²), 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")
```

Multiple metris

