

Buffer test

Contents

0.1 Packages	1
0.2 Landscape buffer	3

0.1 Packages

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

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)
```

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")
```

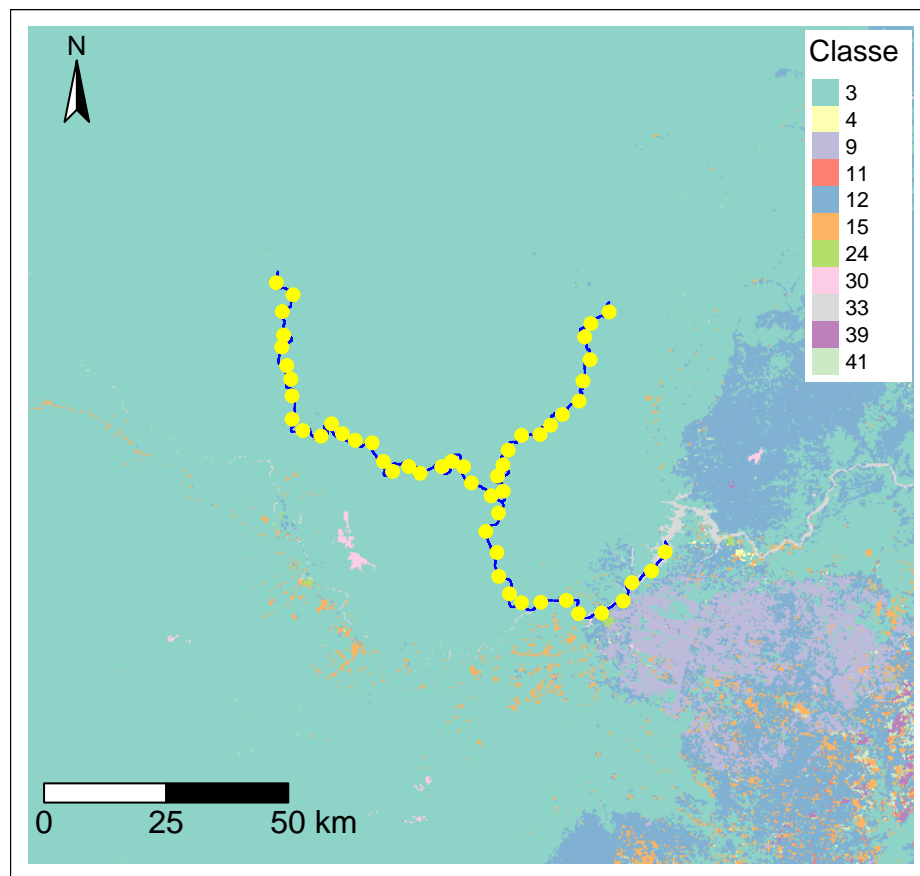


Figure 1: Cobertura da terra ao redor do Rio Araguari em 2020. Mostrando os pontos de amostragem (pontos amarelas) cada 5 quilômetros ao longo do rio.

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

0.2 Landscape buffer

Para amostrar métricas de paisagem dentro de um certo buffer em torno de pontos de amostra, existe a função `sample_lsm()`.

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

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	ca	234.20734	1	99.9608
1	class	1	NA	ca	79.61799	1	99.9608

Calculate proportion

For all

```
minha_amostra_250 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 250, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 250)
minha_amostra_500 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 500, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 500)
minha_amostra_1000 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 1000, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 1000)
minha_amostra_2000 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 2000, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 2000)
minha_amostra_4000 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 4000, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 4000)
minha_amostra_8000 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 8000, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 8000)
minha_amostra_16000 <- sample_lsm(floresta_2020, y = rsm_31976[1, ],
                                size = 16000, shape = "circle",
                                metric = "ca") %>%
  mutate(raio = 16000)
```

join

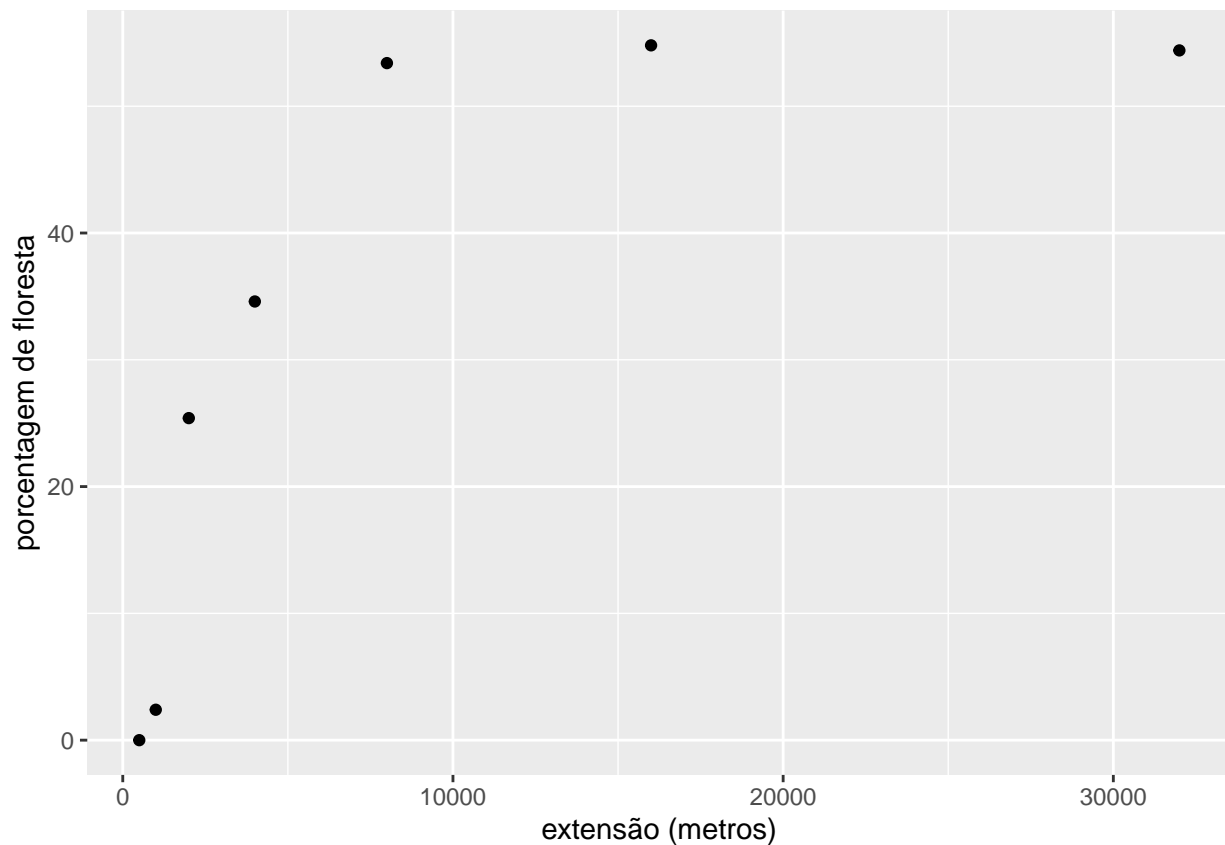
```
bind_rows(minha_amostra_250,
          minha_amostra_500,
          minha_amostra_1000,
          minha_amostra_2000,
          minha_amostra_4000,
          minha_amostra_8000,
          minha_amostra_16000) -> amostras_metricas
```

calculate proportion

```
amostras_metricas %>%
  mutate(floresta_ha = if_else(class == 1, value, 0)) %>%
  group_by(raio) %>%
  summarise(area_total_ha = round(sum(value), 2),
            area_floresta_ha = round(max(floresta_ha), 2)) %>%
  mutate(proporcao_floresta =
    round(area_floresta_ha / area_total_ha, 3),
    porcentagem_floresta =
    round((area_floresta_ha / area_total_ha)*100, 1)) -> amostras_metricas_resumo
```

make a graph

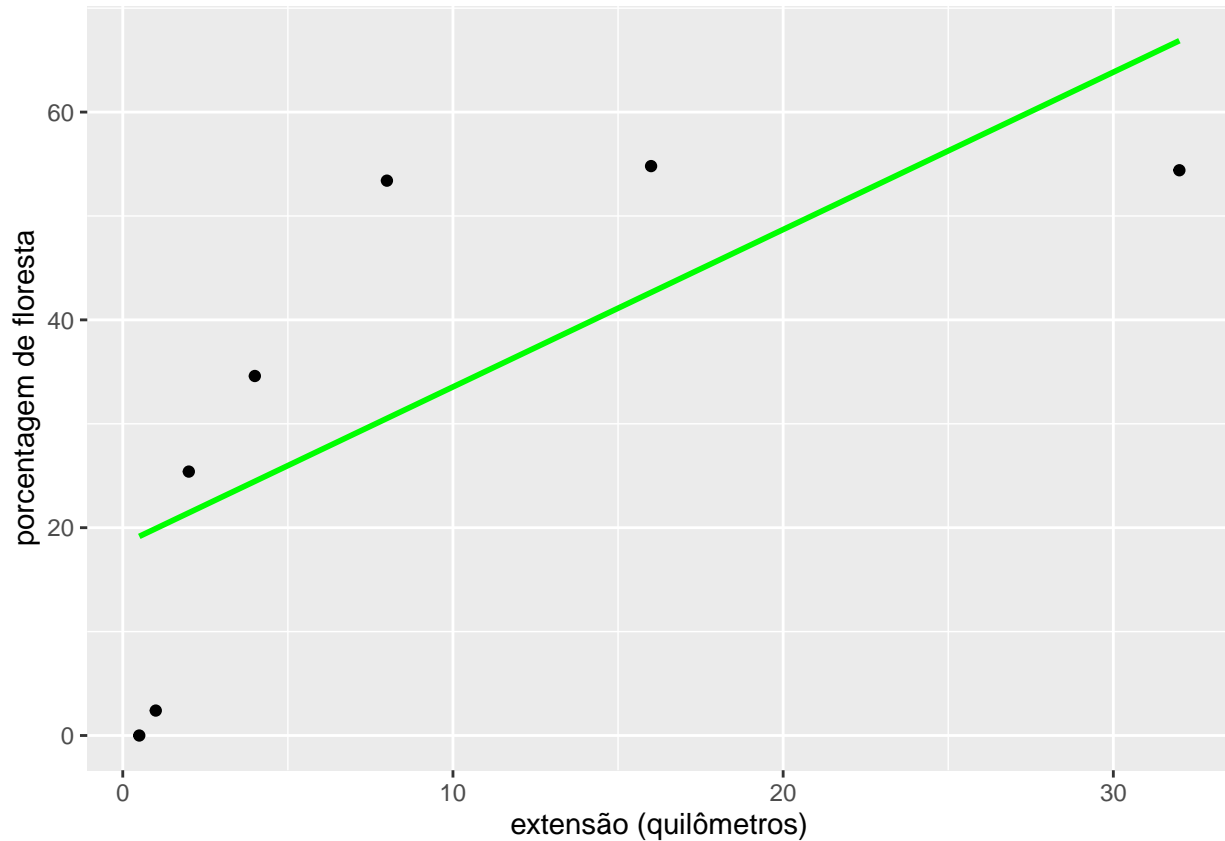
```
amostras_metricas_resumo %>%
  ggplot(aes(x=2*raio, y=porcentagem_floresta)) +
  geom_point() +
  labs(x = "extensão (metros)", y = "porcentagem de floresta")
```



make a better graph

```
amostras_metricas_resumo %>%  
  ggplot(aes(x=(2*raio)/1000, y=porcentagem_floresta)) +  
  geom_point() +  
  stat_smooth(method = "lm", se = FALSE, color = "green") +  
  labs(x = "extensão (quilômetros)", y = "porcentagem de floresta")
```

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



compare linear and non linear

```
amostras_metricas_resumo %>%  
  ggplot(aes(x=(2*raio)/1000, y=porcentagem_floresta)) +  
  geom_point() +  
  stat_smooth(method = "lm", color = "green") +  
  coord_cartesian(ylim = c(0,100)) +  
  labs(title = "Modelo linear",  
       x = "extensão (quilômetros)",  
       y = "porcentagem de floresta") -> fig_lm  
  
amostras_metricas_resumo %>%  
  ggplot(aes(x=(2*raio)/1000, y=porcentagem_floresta)) +  
  geom_point() +  
  stat_smooth(method = "gam", formula = y ~ s(x, k = 5)) +
```

```
coord_cartesian(ylim = c(0,100)) +
labs(title = "Modelo não-linear",
      x = "extensão (quilômetros)",
      y = "porcentagem de floresta") -> fig_gam

grid.arrange(fig_lm, fig_gam, nrow=1)
```

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

