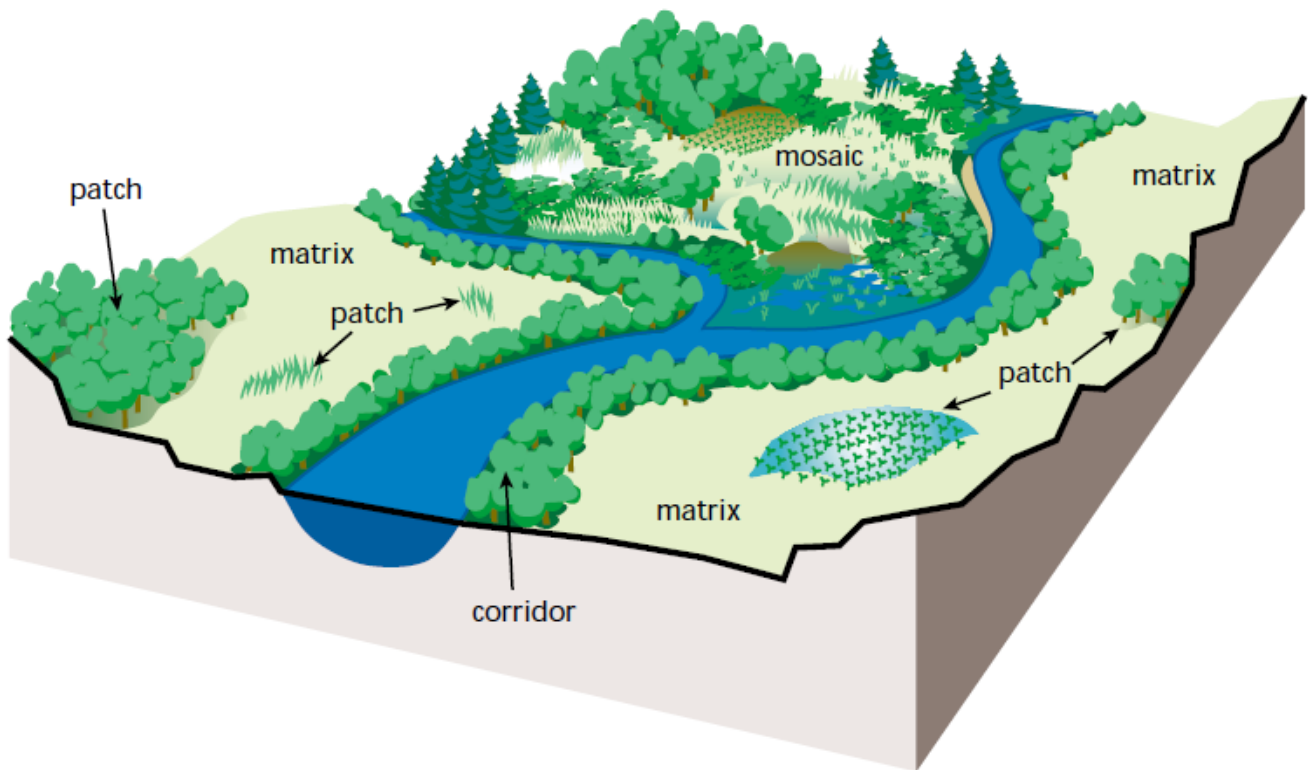


# Praticando Métricas de Paisagem utilizando o R

Msc. Pablo Silva

Universidade Federal de Goiás



## Carregando os pacotes necessários para o nosso estudo de métricas da paisagem

Nessa função, os pacotes que já estão instalados em seu computador serão carregados, e aqueles que ainda requerem instalação serão instalados e carregados.

```
# https://gist.github.com/stevenworthington/3178163

ipak <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}

ipak(c('terra', 'landscapemetrics', 'ggplot2'))

## Carregando pacotes exigidos: terra

## terra 1.7.39

## Carregando pacotes exigidos: landscapemetrics

## Warning: package 'landscapemetrics' was built under R version 4.3.2

## Carregando pacotes exigidos: ggplot2

##           terra landscapemetrics           ggplot2
##           TRUE              TRUE              TRUE
```

## Preparando os dados para a análise

Antes de começar a nossa aventura na análise de métricas da paisagem, primeiro faremos uma preparação dos dados de classificação obtidos pelo MapBiomas coleção 8.0. Os dados que utilizaremos aqui estão listados no repositório dentro da pasta **Dados**. Nesse primeiro momento, listaremos os arquivos contidos em nosso diretório. Veremos que os dados são desde 1985 até 2022.

```
dir <- list.dirs("../Dados")
files <- list.files(dir[2], pattern = ".tif")
```

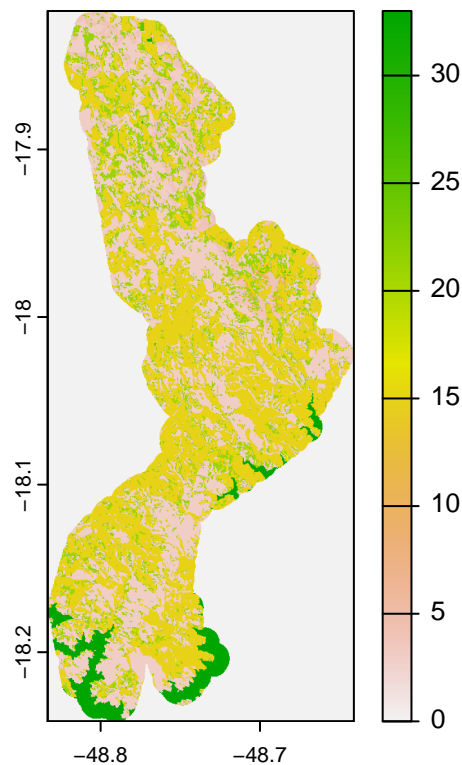
Agora criaremos uma lista para armazenar os dados obtidos pelo MapBiomas. Além disso, rodaremos um código que vai ler cada um dos arquivos e salvar dentro dessa lista.

```
BnoC <- list()

for (i in 1:length(files)) {
  BnoC[[i]] <- rast(paste(dir[2],files[i], sep = "/"))
}; rm(i)
```

Visualizando as classes para o ano de 1985.

```
plot(BnoC[[1]])
```



## Reclassificando a paisagem

Nessa etapa, faremos uma reclassificação dos dados para que seja possível a nossa análise da paisagem. Afinal, quando estamos interessados em uma análise de métricas da paisagem, nem todas as classes de uso e cobertura da terra são de fato necessárias e as classes escolhidas devem estar de acordo com o nosso objetivo e com os requerimentos das espécies estudadas. Nesse exemplo, teremos três classes: (1) natural; (2) antrópico; e (3) água.

Antes de realizar a reclassificação, listei todos os usos em cada um dos anos (1985-2022) e atribuí um valor correspondente de acordo com os objetivos dessa análise.

```
classes_uso <- c(0,3,15,21,4,33,11,12,24,39,25,41,9,20)
classes_paisagem <- c(NA,1,2,2,1,3,1,1,2,2,2,2,2,2)

# Unindo os arquivos em uma matriz
classes <- cbind(classes_uso, classes_paisagem)
```

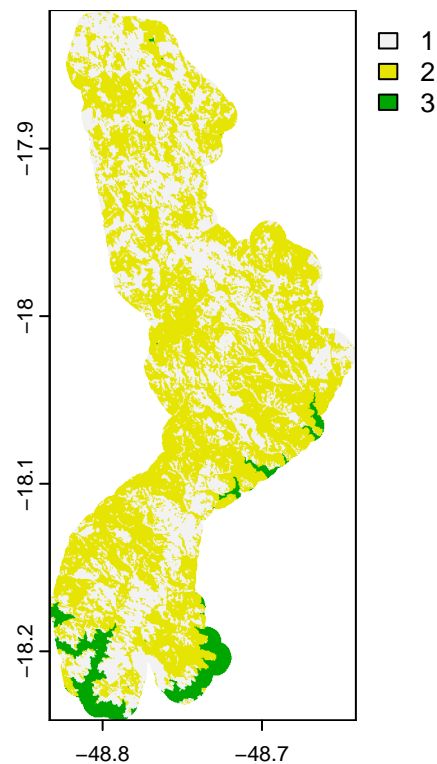
As classes de uso correspondem às classes originais obtidas pelo MapBiomas e as classes da paisagem são as novas categorias.

Abaixo iremos automatizar essa reclassificação.

```
for (i in 1:length(BnoC)) {  
  BnoC[[i]] <- terra::classify(BnoC[[i]],classes)  
}
```

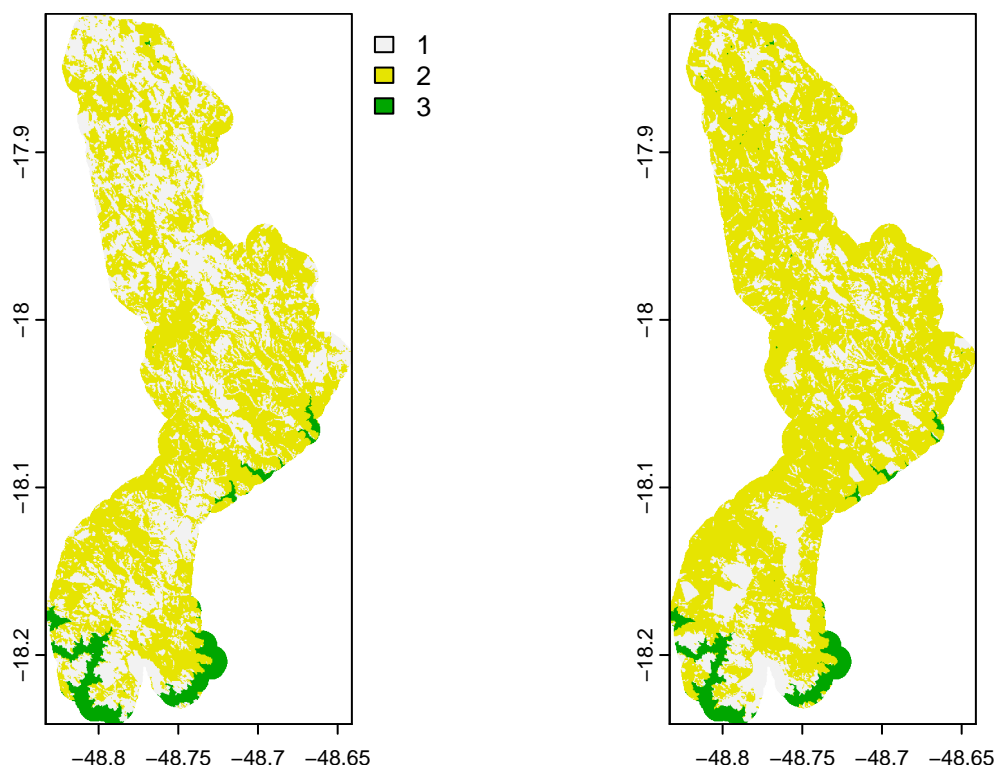
Visualizando as novas classes para o ano de 1985.

```
plot(BnoC[[1]])
```



A partir de agora já podemos explorar nossos dados por meio de métricas da paisagem. No entanto, antes faremos uma análise visual sobre as mudanças no uso e cobertura da terra e observar como essas classes se modificam ao longo do tempo. Observaremos a discrepância entre o ano de 1985 e 2022.

```
par(mfrow = c(1,2))
plot(BnoC[[1]])
plot(BnoC[[38]])
```



## Praticando métricas da paisagem

As métricas da paisagem podem ser analisadas em três níveis diferentes: **mancha**, **classe** e **paisagem**. Cada uma delas respondem perguntas diferentes, mas também podem se complementar e contribuir para uma resposta ecológica mais robusta.

A análise de métricas em nível de **mancha** pode responder questões em termos de estrutura, tal como área, formato e quantidade, enquanto que em nível de **classe** podemos ver a proporção de cada uma das classes analisadas, dentre outras métricas que retornarão como essas classes estão organizadas. Já em nível de **paisagem**, podemos obter respostas relacionadas a estrutura geral, por exemplo, conectividade. Abaixo veremos alguns exemplos para cada um dos níveis.

## Conferindo o status da paisagem

Começaremos a nossa análise de métricas da paisagem conferindo se nossa paisagem está adequada para uso no pacote *landscapemetrics*.

```
check_landscape(BnoC[[38]])
```

```
## Warning: Caution: Coordinate reference system not metric - Units of results
## based on cellsizes and/or distances may be incorrect.
```

```
##   layer      crs   units   class n_classes OK
## 1      1 geographic degrees integer      3   x
```

Após conferir se a paisagem está de acordo, recebemos a mensagem de que o dado espacializado não está em um sistema de coordenada métrico. Isso indica que precisamos projetar nossa paisagem para um sistema de coordenada que satisfaça os requisitos do pacote. Dentre as opções que temos, as projeções do tipo UTM costumam ser bastante utilizadas. No entanto, ela também possui uma limitação que acontece quando o nosso arquivo do tipo raster está presente em duas ou mais UTMs. Nesse caso precisamos apelar para outro tipo de projeção, tal como Albers.

Abaixo observaremos o sistema de coordenadas atribuído em nossa paisagem.

```
BnoC[[38]]
```

```
## class      : SpatRaster
## dimensions  : 1573, 713, 1  (nrow, ncol, nlyr)
## resolution  : 0.0002694946, 0.0002694946  (x, y)
## extent      : -48.83323, -48.64108, -18.24128, -17.81737  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (EPSG:4326)
## source(s)   : memory
## name        : classification_2022
## min value    :                1
## max value    :                3
```

Com isso, a gente consegue observar que o nosso sistema de coordenada está como longitude e latitude WGS 84 (EPSG: 4326). Portanto, iremos projetar as paisagens para o sistema de coordenada UTM. Primeiramente iremos avaliar a zona na em que a nossa paisagem está inserida.

Abaixo segue um código simples que retorna o número da zona.

```
# floor((longitude + 180) / 6) + 1
# Aqui substituiremos os valores mínimos e máximos de longitude na função

floor((-48.83323 + 180) / 6) + 1
```

```
## [1] 22
```

```
floor((-48.64108 + 180) / 6) + 1
```

```
## [1] 22
```

Com essa função, descobrimos que a nossa paisagem está inserida na zona 22 Sul, visto que a paisagem está localizada no estado de Goiás, Brasil.

Nesse caso, faremos a projeção dos nossos arquivos rasters para a projeção UTM 22 Sul. Caso contrário, se a nossa paisagem estivesse localizada em mais de uma zona, faríamos a projeção para um sistema de coordenada global em metros.

```
newcrs <- "+proj=utm +south +zone=22 +datum=WGS84 +units=m +no_defs"
w = project(BnoC[[1]], newcrs)
w
```

```
## class      : SpatRaster
## dimensions  : 1593, 706, 1  (nrow, ncol, nlyr)
## resolution  : 29.62084, 29.62084  (x, y)
## extent      : 729110.5, 750022.8, 7981505, 8028691  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=22 +south +datum=WGS84 +units=m +no_defs
## source(s)   : memory
```



```
## name      : classification_1985
## min value  : 1
## max value  : 3
```

Realizando uma nova conferência em nosso dado.

```
check_landscape(w)
```

```
## Warning: Caution: Land-cover classes must be decoded as integer values.
```

```
## Warning: Caution: More than 30 land cover-classes - Please check if discrete
## land-cover classes are present.
```

```
##   layer      crs units      class n_classes OK
## 1      1 projected      m non-integer    94002  x
```

Veremos que ele ainda não atende aos requisitos do pacote porque por algum motivo ele modificou a classe para non-integer. Corrigiremos isso utilizando a função `round()`.

```
for (i in 1:length(BnoC)) {
  w <- project(BnoC[[i]], newcrs)
  BnoC[[i]] <- round(w)
}
```

Avaliaremos novamente se o dado atende ao requisito.

```
check_landscape(BnoC[[1]])
```

```
##   layer      crs units      class n_classes OK
## 1      1 projected      m integer          3  v
```

Pronto! Nosso dado está pronto para ser utilizado no *landscapemetrics*.

## Métricas em nível de mancha

### Área da mancha (*lsm\_p\_area*)

```
area <- lsm_p_area(BnoC[[38]], directions = 8) # 2022
area
```

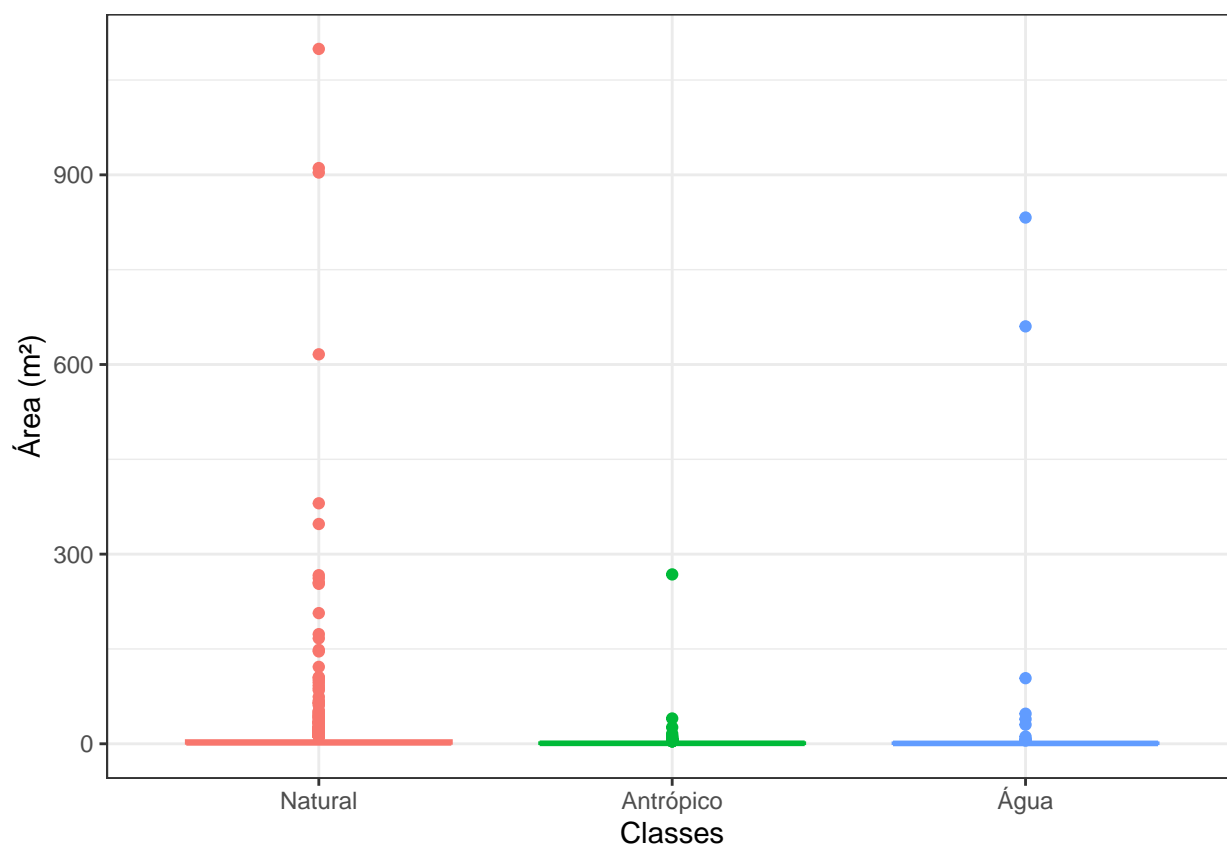
```
## # A tibble: 1,211 x 6
##   layer level class    id metric  value
##   <int> <chr> <int> <int> <chr>   <dbl>
## 1     1   1 patch     1     1 area    2.46
## 2     1   1 patch     1     2 area   167.
## 3     1   1 patch     1     3 area    2.02
## 4     1   1 patch     1     4 area   26.5
## 5     1   1 patch     1     5 area   15.2
## 6     1   1 patch     1     6 area   10.6
## 7     1   1 patch     1     7 area    6.32
## 8     1   1 patch     1     8 area   101.
## 9     1   1 patch     1     9 area    6.40
## 10    1   1 patch     1    10 area    0.526
## # i 1,201 more rows
```

Vamos observar a diferença no tamanho da área entre as classes. Para isso, faremos um box plot que vai destacar melhor a variação no tamanho das áreas.

```
area <- area[-931,] # removendo ponto discrepante

ggplot(data = area,
       aes(x = as.factor(class),
           y = value,
           colour = as.factor(class))) +
  geom_boxplot() +
```

```
scale_x_discrete(breaks = c(1,2,3),
                 labels = c("Natural", "Antrópico", "Água")) +
xlab("Classes") +
ylab("Área (m²)") +
theme_bw() + theme(legend.position = "none")
```



### Porcentagem da área do interior (*lsm\_p\_cai*)

```
cai <- lsm_p_cai(BnoC[[38]],
                directions = 8,
                consider_boundary = FALSE,
                edge_depth = 1)
cai
```

```
## # A tibble: 1,211 x 6
```

```
##   layer level class    id metric value
##   <int> <chr> <int> <int> <chr>  <dbl>
##  1     1 patch     1     1 cai     25
##  2     1 patch     1     2 cai    85.9
##  3     1 patch     1     3 cai    43.5
##  4     1 patch     1     4 cai    64.6
##  5     1 patch     1     5 cai    43.4
##  6     1 patch     1     6 cai    64.5
##  7     1 patch     1     7 cai    30.6
##  8     1 patch     1     8 cai    89.4
##  9     1 patch     1     9 cai    63.0
## 10     1 patch     1    10 cai     0
## # i 1,201 more rows
```

Nessa análise a variável resposta corresponde a porcentagem de área de interior existe em cada uma das manchas analisadas, sendo 0 a indicação de que mancha é totalmente borda.

```
rm(cai)
```

### Área do interior (*lsm\_p\_core*)

Essa métrica calcula a área do interior, sendo a área maior que zero.

```
core <- lsm_p_core(BnoC[[38]],
                  directions = 8,
                  consider_boundary = FALSE,
                  edge_depth = 1)
core
```

```
## # A tibble: 1,211 x 6
##   layer level class    id metric  value
##   <int> <chr> <int> <int> <chr>   <dbl>
##  1     1 patch     1     1 core   0.614
```

```
## 2      1 patch      1      2 core    143.
## 3      1 patch      1      3 core     0.877
## 4      1 patch      1      4 core    17.1
## 5      1 patch      1      5 core     6.58
## 6      1 patch      1      6 core     6.84
## 7      1 patch      1      7 core     1.93
## 8      1 patch      1      8 core    90.1
## 9      1 patch      1      9 core     4.04
## 10     1 patch      1     10 core      0
## # i 1,201 more rows
```

A variável resposta está na unidade de medida hectare.

### Circularidade da mancha (*lsm\_p\_circle*)

Esta métrica é relacionada com o formato da mancha, indicando a circularidade de cada mancha.

```
circle <- lsm_p_circle(BnoC[[38]], directions = 8)
circle
```

```
## # A tibble: 1,211 x 6
##   layer level class    id metric value
##   <int> <chr> <int> <int> <chr>  <dbl>
## 1      1 patch      1      1 circle 0.554
## 2      1 patch      1      2 circle 0.531
## 3      1 patch      1      3 circle 0.349
## 4      1 patch      1      4 circle 0.634
## 5      1 patch      1      5 circle 0.689
## 6      1 patch      1      6 circle 0.654
## 7      1 patch      1      7 circle 0.579
## 8      1 patch      1      8 circle 0.485
```

```
## 9      1 patch      1      9 circle 0.378
## 10     1 patch      1     10 circle 0.794
## # i 1,201 more rows
```

Esse índice varia de 0 a 1, sendo 0 manchas completamente circulares e 1 manchas lineares.

### **Distância Euclidiana do Vizinho mais Próximo (*lsm\_p\_enn*)**

Essa é uma métrica de agregação, indicando a distância mais próxima entre outra mancha da mesma classe.

```
ennm <- lsm_p_enn(BnoC[[38]], directions = 8, verbose = TRUE)
ennm
```

```
## # A tibble: 1,211 x 6
##   layer level class    id metric value
##   <int> <chr> <int> <int> <chr>  <dbl>
## 1      1 patch      1      1 enn    608.
## 2      1 patch      1      2 enn    88.9
## 3      1 patch      1      3 enn    88.9
## 4      1 patch      1      4 enn    83.8
## 5      1 patch      1      5 enn    66.2
## 6      1 patch      1      6 enn   107.
## 7      1 patch      1      7 enn   160.
## 8      1 patch      1      8 enn    66.2
## 9      1 patch      1      9 enn    83.8
## 10     1 patch      1     10 enn   107.
## # i 1,201 more rows
```

A variável resposta está em metros.

## Razão Perímetro-Área (*lsm\_p\_para*)

```
para <- lsm_p_para(BnoC[[38]], directions = 8)
```

```
para
```

```
## # A tibble: 1,211 x 6
##   layer level class    id metric  value
##   <int> <chr> <int> <int> <chr>   <dbl>
## 1     1     1 patch     1     1 para  0.0410
## 2     1     1 patch     1     2 para  0.00689
## 3     1     1 patch     1     3 para  0.0323
## 4     1     1 patch     1     4 para  0.0183
## 5     1     1 patch     1     5 para  0.0301
## 6     1     1 patch     1     6 para  0.0179
## 7     1     1 patch     1     7 para  0.0384
## 8     1     1 patch     1     8 para  0.00541
## 9     1     1 patch     1     9 para  0.0194
## 10    1     1 patch     1    10 para  0.0788
## # i 1,201 more rows
```

Essa análise não possui unidade de medida.

## Métricas em nível de classe

### Índice de agregação (*lsm\_c\_ai*)

Essa métrica calcula a porcentagem de agregação para cada uma das classes.

```
## # A tibble: 3 x 6
##   layer level class    id metric value
##   <int> <chr> <int> <int> <chr>   <dbl>
## 1     1     1 class     1    NA ai    87.6
```

```
## 2      1 class      2      NA ai      95.3
## 3      1 class      3      NA ai      95.3
```

### Média do tamanho da área (*lsm\_c\_area\_mn*)

Essa métrica calcula a média do tamanho da área para cada uma das classes.

```
area_mn <- lsm_c_area_mn(BnoC[[38]], directions = 8)
area_mn
```

```
## # A tibble: 3 x 6
##   layer level class    id metric  value
##   <int> <chr> <int> <int> <chr>   <dbl>
## 1      1 class      1      NA area_mn  12.2
## 2      1 class      2      NA area_mn 141.
## 3      1 class      3      NA area_mn  34.7
```

A variável resposta é em hectare.

### Desvio padrão do tamanho da área (*lsm\_c\_area\_sd*)

Essa métrica calcula o desvio padrão da área de cada mancha.

```
area_sd <- lsm_c_area_sd(BnoC[[38]], directions = 8)
area_sd
```

```
## # A tibble: 3 x 6
##   layer level class    id metric  value
##   <int> <chr> <int> <int> <chr>   <dbl>
## 1      1 class      1      NA area_sd   65.2
## 2      1 class      2      NA area_sd 2100.
## 3      1 class      3      NA area_sd  147.
```

A variável resposta é em hectare.



## Número de manchas (*lsm\_c\_np*)

Essa métrica calcula o número de manchas pertencentes a cada uma das classes.

```
np <- lsm_c_np(BnoC[[38]], directions = 8)
```

```
np
```

```
## # A tibble: 3 x 6
##   layer level class    id metric value
##   <int> <chr> <int> <int> <chr>   <dbl>
## 1     1    1 class     1    NA np      930
## 2     1    1 class     2    NA np      230
## 3     1    1 class     3    NA np       51
```

Vamos brincar um pouco com essa métrica e observar o número de fragmentos ao longo do tempo (1985-2022)

```
dado <- as.data.frame(matrix(data = NA,
                             nrow = length(BnoC),
                             ncol = 2))
```

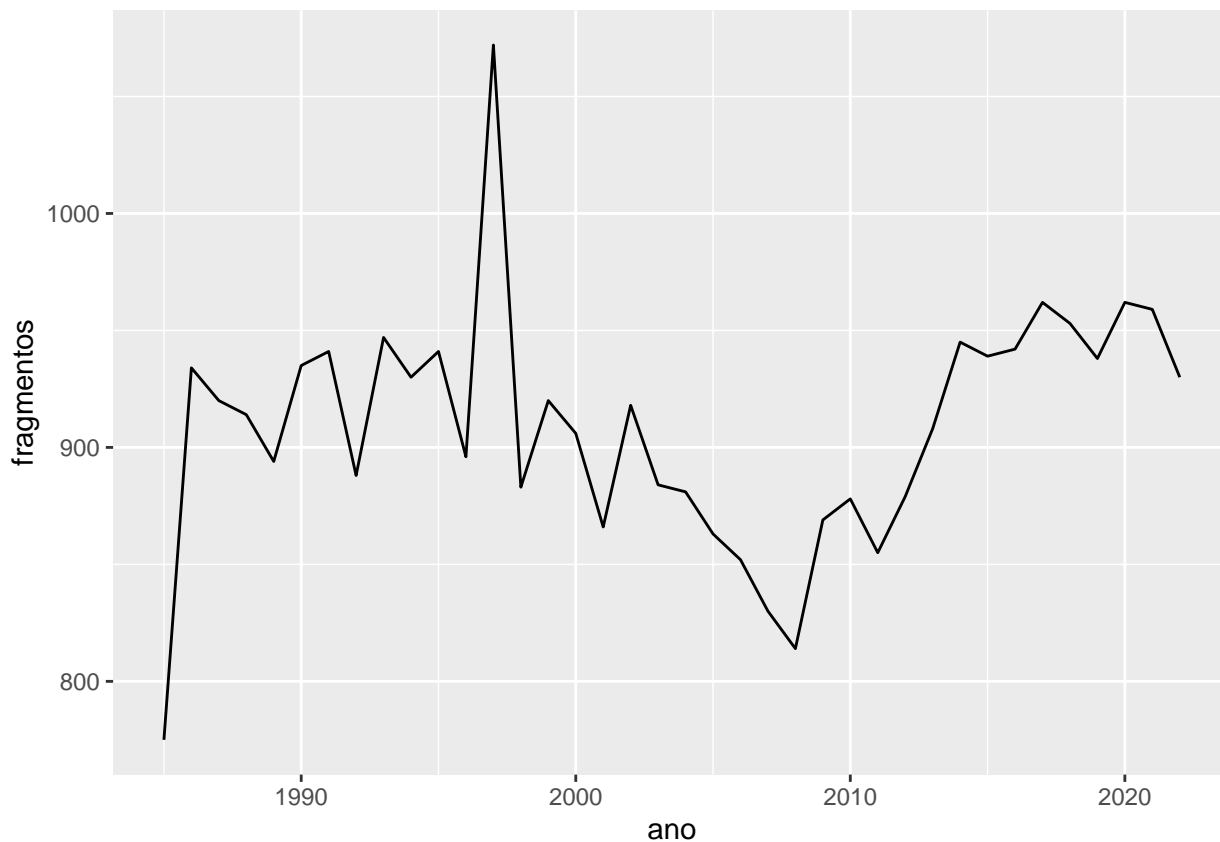
```
for (i in 1:length(BnoC)) {
  anos <- seq(from = 1985, to = 2022, by = 1)
  np <- lsm_c_np(BnoC[[i]], directions = 8)
  dado[i,1] <- anos[i]
  dado[i,2] <- np[1,6]
}
```

```
colnames(dado) <- c("ano", "fragmentos")
head(dado)
```

```
##   ano fragmentos
## 1 1985         775
```

```
## 2 1986      934
## 3 1987      920
## 4 1988      914
## 5 1989      894
## 6 1990      935
```

```
ggplot(data = dado, aes(x = ano, y = fragmentos)) +
  geom_line()
```



### Porcentagem da paisagem (*lsm\_c\_pland*)

Essa métrica calcula a porcentagem de cada uma das classes na paisagem.

```
## # A tibble: 3 x 6
##   layer level class    id metric value
##   <int> <chr> <int> <int> <chr>   <dbl>
```

```
## 1      1 class      1      NA pland  24.8
## 2      1 class      2      NA pland  71.3
## 3      1 class      3      NA pland   3.88
```

## Métricas em nível de paisagem

### Número de manchas (*lsm\_l\_np*)

Essa métrica calcula a porcentagem de agregação dentro da paisagem.o número de manchas totais na paisagem

```
np <- lsm_l_np(BnoC[[38]], directions = 8)
np
```

```
## # A tibble: 1 x 6
##   layer level      class    id metric value
##   <int> <chr>      <int> <int> <chr>  <dbl>
## 1      1 landscape    NA     NA np      1211
```

### Índice de diversidade de Simpson (*lsm\_l\_sidi*)

```
sidi <- lsm_l_sidi(BnoC[[38]], directions = 8)
sidi
```

```
## # A tibble: 1 x 6
##   layer level      class    id metric value
##   <int> <chr>      <int> <int> <chr>  <dbl>
## 1      1 landscape    NA     NA sidi   0.429
```

O resultado varia entre 0 e 1, mas não possui unidade de medida.

## Área total da paisagem (*lsm\_l\_ta*)

Essa métrica calcula a área total da paisagem.

```
ta <- lsm_l_ta(BnoC[[38]], directions = 8)
```

```
ta
```

```
## # A tibble: 1 x 6
```

```
##   layer level      class    id metric  value
```

```
##   <int> <chr>      <int> <int> <chr>   <dbl>
```

```
## 1      1 landscape    NA    NA ta     45604.
```

A variável resposta está em hectare.