

# Influência da APP na produção de madeira na floresta Amazônica

Leonardo C. Peres

2023-05-12

---

## Apresentação

Neste projeto o objetivo foi utilizar a linguagem **R** para analisar a influência da presença de APP na produção em m<sup>3</sup> de madeira e da quantidade de indivíduos presentes em diferentes Unidades de Trabalho (UT) de manejo florestal realizado na floresta Amazônica.

---

## Hipotese

- **H0** A presença de APP nas UTs exerce influência na produção de madeira e quantidade de indivíduos;
  - **H1** A presença de APP nas UTs não exerce influência significativa na produção de madeira e quantidade de indivíduos;
- 

## Etapas

Para realização do projeto o mesmo foi dividido em três etapas:

- **Análise exploratória** Estudo dos dados e possibilidades de trabalhos
  - **Modelagem** Análise estatísticas para responder a hipótese
  - **Rmarkdown** Apresentação do projeto e dos códigos utilizados nas etapas anteriores
- 

## Dados

Os dados utilizados são fornecidos pelo projeto **FlorestaR** disponibilizados em um diretório publico do GitHub

O projeto **FlorestaR** reúne professores, estudantes e profissionais dedicados à redação da série **FlorestaR** de livros sobre o uso do **R** como ferramenta de análise de dados florestais.

Para mais informações a respeito do projeto acesse: [github.com/LuizEstraviz/FlorestaR\\_dados](https://github.com/LuizEstraviz/FlorestaR_dados)

---

## 1. Exploratória

Nesta etapa do projeto, o objetivo foi verificar qual a relação existente entre a presença de APP na UT e a produção madeireira assim como o número de indivíduos presente.

Para isso foi criado um data-frame shapefile contendo as 10 UTs, e suas correspondentes: (quantidade de indivíduos por hectare, volume total por hectare, volume médio, área de APP dentro da UT e área da UT). Com isso foram plotados gráficos afim de representar as relações existentes entre a área de APP

---

## Abrindo os dados

```
remove(list=ls())

#Definindo o diretório onde os arquivos estão armazenados
gitOnde <- "https://github.com/FlorestaR/dados/blob/main/1_AMAZON"
gitNome <- "shapes.zip"
gitArqv <- file.path(gitOnde, gitNome) %>% paste0("?raw=true")

#Criando diretório temporário e descompactando pasta zipada
tmpd <- tempdir(check = TRUE) # diretório temporário
zipf <- file.path(tmpd, "shapes.zip") # arquivo temporário

if(!file.exists(zipf)) # garante download de dados binários (wb)
  download.file(gitArqv, mode="wb", destfile = zipf)

unzip(zipf, exdir = tmpd) # shape é unzipped no diretório temporário
unlink(zipf) # deleta o arquivo zipado

# leitura das camadas
dadosArv <- paste0(tmpd, "/shapes/Arvores.shp") %>% read_sf()
dadosUts <- paste0(tmpd, "/shapes/UTs.shp") %>% read_sf()
dadosApp <- paste0(tmpd, "/shapes/APP.shp") %>% read_sf()
```

---

## Gerando o dataframe com os valores de volume e número de indivíduos por UT

```
#filtrando meus registros
registros <- dadosArv %>%
  filter(destino == "PAB" | destino == "CF") %>%
  select(codARV, altura, dap, volume, codUT)

# Criar um dataframe com todas as codUT
todos_codUTs <- data.frame(unique(dadosUts$UT))
colnames(todos_codUTs) <- c("codUT")

#Gerando a média, a soma dos volumes e nº de espécies por UT
mediaVolUts <- merge(todos_codUTs, aggregate(ifelse(is.na(volume), 0, volume) ~ codUT,
  data = registros, mean), by = "codUT", all.x = TRUE)#media
somaVolUts <- merge(todos_codUTs, aggregate(ifelse(is.na(volume), 0, volume) ~ codUT,
  data = registros, sum), by = "codUT", all.x = TRUE)#Soma
contagemVolUts <- merge(todos_codUTs, aggregate(ifelse(is.na(volume), 0, volume) ~ codUT,
  data = registros, FUN = length), by = "codUT", all.x = TRUE)#Qtd

#Unindo todos os valores em um dataframe apenas e renomeado as colunas
resumoUts <- merge(merge(mediaVolUts, somaVolUts, by = "codUT"), contagemVolUts, by =
  "codUT")
```

```
colnames(resumoUts) <- c("codUT", "mediaVolume", "somaVolume", "numeroIndividuos")
resumoUts[is.na(resumoUts)] <- 0
```

## Calculando a área de APP existente dentro de cada UT

```
# Dividir o shp dadosAPP em feições por Uts -> APP por Uts
poligono_recortado <- st_intersection(dadosApp, dadosUts)

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries

appUts <- st_cast(poligono_recortado, "MULTIPOLYGON")

#Calcular a área da app nas Uts
area <- round(st_area(appUts)/10000,3)

#Removendo a unidade de medida do texto e transformando em número
appUts$AreaAPP <- as.numeric(gsub("[m^]", "", as.character(area)))

#Agrupando os valores de área da app pelas Uts e renomeado as colunas
areaAppUts <- aggregate(AreaAPP ~ UT, data = appUts, sum)
colnames(areaAppUts) <- c("codUT", "areaAPP")

#Enviando os dados da tabela areaAppUts para o shp resumoUts através d ID
resumoUts <- merge(resumoUts, areaAppUts, by = "codUT")

#Inserindo o valor bruto de área das Uts
resumoUts$area <- 100.00

#calculando a coluna de área líquida area com app - area da app
resumoUts$areaSApp <- resumoUts$area - resumoUts$areaAPP

#Selecionando apenas as colunas que interessa
resumoUts <- resumoUts %>%
  select(codUT,mediaVolume,somaVolume,numeroIndividuos,areaAPP,area,areaSApp)

#Criando as colunas com as informações de volume/ha e individuos/ha
resumoUts$volumeHectare <- resumoUts$somaVolume / resumoUts$areaSApp
resumoUts$individuosHectare <- resumoUts$numeroIndividuos / resumoUts$areaSApp

head(resumoUts)

##   codUT mediaVolume somaVolume numeroIndividuos areaAPP area areaSApp
## 1     1    2.964085    4209.00           1420    5.054 100    94.946
## 2     2    3.141168    3926.46           1250   12.115 100    87.885
## 3     3    2.974856    3308.04           1112   18.005 100    81.995
## 4     4    2.965656    3502.44           1181   10.937 100    89.063
## 5     5    3.065453    3316.82           1082   11.044 100    88.956
## 6     6    3.164379    3237.16           1023   27.144 100    72.856
##   volumeHectare individuosHectare
## 1    44.33046         14.95587
## 2    44.67725         14.22313
## 3    40.34441         13.56180
```

## 4	39.32542	13.26028
## 5	37.28607	12.16332
## 6	44.43230	14.04140

---

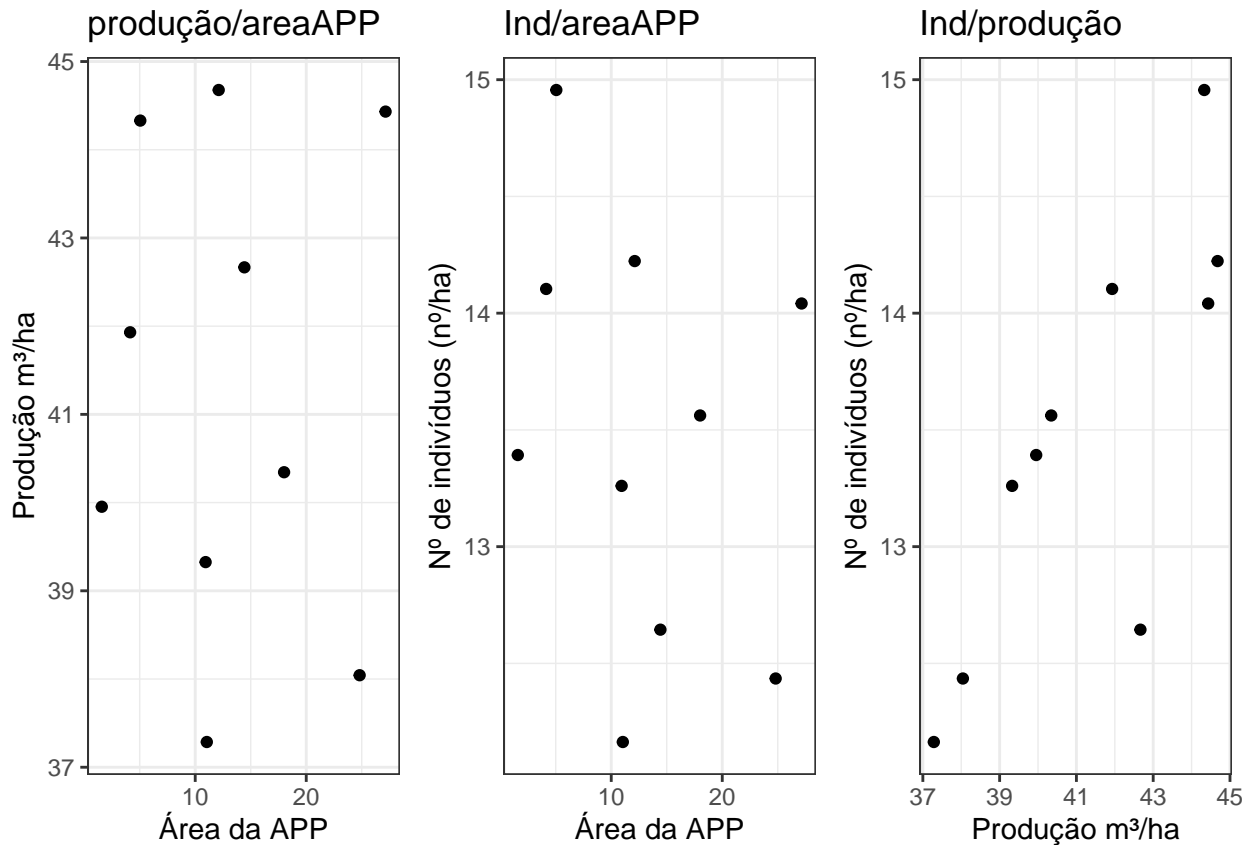
## Plotagem dos gráficos de dispersão

```
GraficoVolume<-ggplot()+
  ggtitle("produção/areaAPP")+
  geom_point(data = resumoUts,
             aes(x= areaAPP, y = volumeHectare))+
  theme_bw()+
  xlab("Área da APP") + ylab("Produção m³/ha")

GraficoIndividuos<-ggplot()+
  ggtitle("Ind/areaAPP")+
  geom_point(data = resumoUts,
             aes(x= areaAPP, y = individuosHectare))+
  theme_bw()+
  xlab("Área da APP") + ylab("Nº de indivíduos (nº/ha)")

GraficoIndVol<-ggplot()+
  ggtitle("Ind/produção")+
  geom_point(data = resumoUts,
             aes(x= volumeHectare, y = individuosHectare))+
  theme_bw()+
  xlab("Produção m³/ha") + ylab("Nº de indivíduos (nº/ha)")

grid.arrange(GraficoVolume, GraficoIndividuos,GraficoIndVol, ncol = 3,nrow = 1)
```



Ao analisar os gráficos percebe-se que não existe relação entre os agentes avaliados (APP X Produção) e (APP X Nº de indivíduos). Sendo assim foram gerados dois mapas para entender melhor sobre a dinâmica dos parâmetros avaliados e sua UT correspondente

## Mapas de localização dos blocos e seus respectivos valores de volume m³ por ha e quantidade de indivíduos

```
colnames(resumoUts) <- c("UT", "mediaVolume", "somaVolume", "numeroIndividuos",
  "areaAPP", "area", "areaSApp", "volumeHectare", "individuosHectare")
```

```
dadosUts <- merge(dadosUts, resumoUts, by = "UT")
```

```
MapaVolUt <- ggplot() +
  ggtitle("Produção m³/ha")+
  geom_sf(data = dadosUts, aes(fill = volumeHectare), colour = "red") +
  scale_fill_gradient(low = "blue", high = "orange") +
  geom_sf_text(data = dadosUts, aes(label = UT), size = 3) +
  geom_sf(data = dadosApp, colour = "red", fill='white', alpha = 0.5) +
  coord_sf(datum=st_crs(29190)) + # Especifica sistema de coord.
  scale_x_continuous(breaks = seq(from = 218500, to = 223000, by = 1000))+
  labs(fill = "Produção m³/ha")
```

```
MapaIndlUt <- ggplot() +
  ggtitle("Individuos/ha")+
  geom_sf(data = dadosUts, aes(fill = individuosHectare), colour = "red") +
```

```

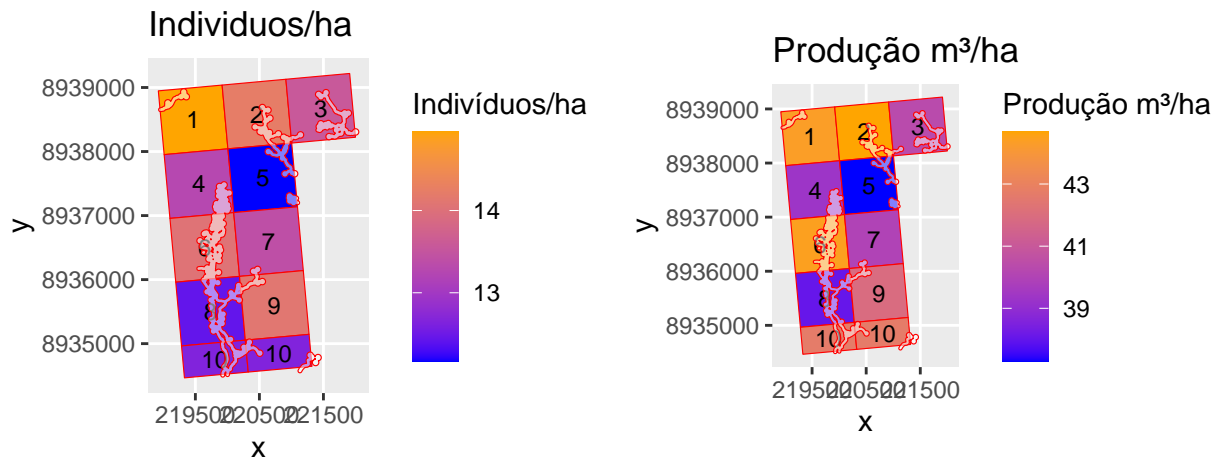
scale_fill_gradient(low = "blue", high = "orange",) +
geom_sf_text(data = dadosUts, aes(label = UT), size = 3) +
geom_sf(data = dadosApp, colour = "red", fill='white', alpha = 0.5) +
coord_sf(datum=st_crs(29190)) + # Especifica sistema de coord.
scale_x_continuous(breaks = seq(from = 218500, to = 223000, by = 1000))+
labs(fill = "Indivíduos/ha")

```

```

grid.arrange(MapaIndlUt, MapaVolUt, ncol = 2,nrow = 1)

```



## 2. Modelagem

Diante dos resultados obtidos durante a etapa exploratória dos dados, optou-se por agrupar as UTs em 3 classes diferentes de acordo com a área de APP dentro de cada uma delas. Foram consideradas UTs com alta influência de APP aquelas que apresentaram área de APP maior que 15 hectares, média influência aquelas que a área da APP se encontrasse entre 7 e 15 hectares e as demais foram consideradas com baixa influência.

### Criação da coluna categorizando as UTs de acordo com a área de APP dentro delas

```

# Criar a nova coluna com base na condição
dadosUts$catAreaApp <- ifelse(dadosUts$areaAPP > 15, 2, ifelse(dadosUts$areaAPP > 7, 1,
0))

```

```

print(dadosUts)

```

```

## Simple feature collection with 11 features and 14 fields
## Geometry type: POLYGON
## Dimension: XY
## Bounding box: xmin: 218921.9 ymin: 8934466 xmax: 222000.2 ymax: 8939219
## Projected CRS: SAD69 / UTM zone 20S
## First 10 features:
##   UT Y_INDEX X_INDEX Area Hectares mediaVolume somaVolume
## 1 1 1 8 1000000.0 100.00000 2.964085 4209.00
## 2 2 1 9 1000000.0 100.00000 3.141168 3926.46
## 3 3 1 10 1000000.0 100.00000 2.974856 3308.04
## 4 4 2 8 1000000.0 100.00000 2.965656 3502.44

```

```

## 5 5 2 9 1000000.0 100.00000 3.065453 3316.82
## 6 6 3 8 1000000.0 100.00000 3.164379 3237.16
## 7 7 3 9 1000000.0 100.00000 2.983255 3931.93
## 8 8 4 8 1000000.0 100.00000 3.059262 2860.41
## 9 9 4 9 1000000.0 100.00000 2.972996 4019.49
## 10 10 5 8 500313.1 50.03131 3.374362 3651.06
## numeroIndividuos areaAPP area areaSApp volumeHectare individuosHectare
## 1 1420 5.054 100 94.946 44.33046 14.95587
## 2 1250 12.115 100 87.885 44.67725 14.22313
## 3 1112 18.005 100 81.995 40.34441 13.56180
## 4 1181 10.937 100 89.063 39.32542 13.26028
## 5 1082 11.044 100 88.956 37.28607 12.16332
## 6 1023 27.144 100 72.856 44.43230 14.04140
## 7 1318 1.586 100 98.414 39.95295 13.39240
## 8 935 24.811 100 75.189 38.04293 12.43533
## 9 1352 4.139 100 95.861 41.93040 14.10375
## 10 1082 14.429 100 85.571 42.66703 12.64447
## geometry catAreaApp
## 1 POLYGON ((220008.4 8938042,... 0
## 2 POLYGON ((221004.3 8938133,... 1
## 3 POLYGON ((221004.3 8938133,... 2
## 4 POLYGON ((220098.9 8937046,... 1
## 5 POLYGON ((221094.8 8937137,... 1
## 6 POLYGON ((220189.5 8936050,... 2
## 7 POLYGON ((221185.4 8936141,... 0
## 8 POLYGON ((220280 8935055, 2... 2
## 9 POLYGON ((221275.9 8935145,... 0
## 10 POLYGON ((220325.3 8934556,... 1

```

## Geração do mapa com as UTs classificadas

```

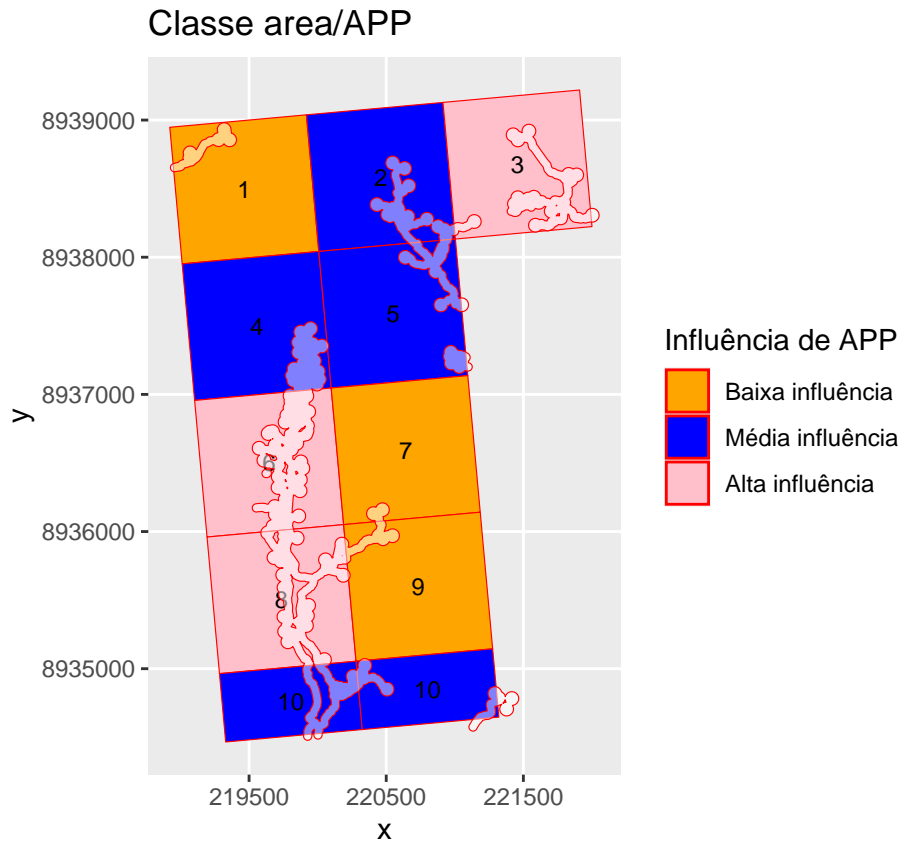
# Converter a variável catAreaApp para o tipo de dados factor
dadosUts$catAreaApp <- factor(dadosUts$catAreaApp)

# Definir cores para cada valor da variável catAreaApp
cores <- c("orange", "blue", "pink")
nomes_cores <- setNames(cores, levels(dadosUts$catAreaApp))

# Plotar o gráfico com cores por valor
MapaClasseAPP <- ggplot() +
  ggtitle("Classe area/APP") +
  geom_sf(data = dadosUts, aes(fill = catAreaApp), colour = "red") +
  geom_sf_text(data = dadosUts, aes(label = UT), size = 3) +
  geom_sf(data = dadosApp, colour = "red", fill = 'white', alpha = 0.5) +
  coord_sf(datum = st_crs(29190)) + # Especifica sistema de coord.
  scale_x_continuous(breaks = seq(from = 218500, to = 223000, by = 1000)) +
  scale_fill_manual(values = nomes_cores, labels = c("Baixa influência",
    "Média influência", "Alta influência")) +
  labs(fill = "Influência de APP") # Altera o título da legenda

# Visualizar o gráfico
print(MapaClasseAPP)

```



## ANOVA

Em seguida, realizou-se a ANOVA para verificar a existência de diferença significativa entre as categorias criadas em relação a produção de madeira (Tabela 1) e ao número de indivíduos (Tabela 2), considerando um nível de significância de 0,05.

## ANOVA Volume

```

modeloAnovaVol <- aov(mediaVolume ~ catAreaApp, data = dadosUts)

# Resumir os resultados da ANOVA
resultadoAnovaVol <- summary(modeloAnovaVol)

resultadoVolTexto <- capture.output(print(resultadoAnovaVol))

# Visualizar o resultado
print(resultadoVolTexto)

## [1] "          Df Sum Sq Mean Sq F value Pr(>F) "
## [2] "catAreaApp  2 0.0866 0.04330  2.246  0.168"
## [3] "Residuals   8 0.1542 0.01928          "

```



## ANOVA nº indivíduos

```
modeloAnovaInd <- aov(individuosHectare ~ catAreaApp, data = dadosUts)

# Resumir os resultados da ANOVA
resultadoAnovaInd <- summary(modeloAnovaInd)

resultadoIndTexto <- capture.output(print(resultadoAnovaInd))

# Visualizar o resultado
print(resultadoIndTexto)
```

```
## [1] "          Df Sum Sq Mean Sq F value Pr(>F)"
## [2] "catAreaApp  2  2.551  1.2757  2.001  0.197"
## [3] "Residuals  8  5.101  0.6376          "
```

Ambas as ANOVAS indicam que os dois parâmetros testados não apresentam diferença significativa entre eles e a presença de APP nas UTs, pois o  $Pr(>f)$  possuem valores maiores que 0,05.

---

## 3. Conclusão

De acordo com os resultados apresentados neste trabalho, a presença da APP nas unidades de trabalhos não são suficientes para gerarem alguma influência sobre a produção de madeira e a quantidade de indivíduos presentes, em um contexto geral considerando todos os indivíduos com DAP maior que 40.

Uma possibilidade seria replicar a mesma sequência lógica que gerou este resultado, porém filtrando o data-frame por espécies, com o objetivo de localizar espécies com maior aptidão a regiões próximas a APPs e outras com menor.

---

RStudioCloud - PRODAPPMANJ