Explorando fatores que afetam a transmissão de COVID-19 na Bahia

Grupo de Estudos em Ecologia Espacial - UFBA

4/17/2020

Apresentação

Esse documento traz os resultados e códigos utilizados nas análises dos fatores que contribuem para a expansão dos casos de covid-19 no estado da Bahia. Esse estudo foi realizado pelo grupo de estudos em ecologia espacial da UFBA, incluindo diversos pesquisadores e laboratórios do Instituo de Biologia.

Pacotes

Se for seguir o código para recriar as análises, antes de inciar, carregue e instale os seguintes pacotes.

```
library(coronabr) # pode baixar aqui: https://github.com/liibre/coronabr
library(tidyverse)
library(car)
library(randomForest)
library(rgdal) #Load map
library(sp) #plot maps
```

Baixar os dados para a Bahia

Com o código abaixo podemos baixar os dados para todos os municipios Bahia. Para saber mais sobre as fontes do dados acesse o seguinte link: https://github.com/liibre/coronabr (https://github.com/liibre/coronabr).

```
covid <- as_tibble(get_corona_br(uf = "BA"))
```

Pequenos ajustes na tabela:

```
covid <- covid %>%
  filter(place_type == "city") %>%
  mutate(city = factor(city, levels = unique(city)))
```

Dados por municipio:

```
mun_covid <- covid %>%
  filter(date == date[1]) %>%
  mutate(afetados = ifelse(confirmed > 0, 1, 0))
```

Estatísticas dos casos na Bahia:

```
mun_covid %>%
  summarise("Casos totais" = sum(confirmed),
            "Mortes totais" = sum(deaths),
            "Número de municipios afetados" = sum(confirmed > 0))
```

Casos totais <int>	Mortes totais <int>	Número de municipios afetados <int>
1504	48	100

1 row

Causas que afetam a ocorrência de COVID-19

Carregando os dados do IBGE e mapbiomas:

```
ibge <- read_csv2(file("D:/Covid/Covid-19_Bahia/new_ibge.csv", encoding="UTF-8")) %
>%
  separate(cidade, c("Cidade", "Estado"), sep = "\\(") %>%
  mutate(Estado = str_remove(Estado, "\\)")) %>%
   filter(Estado == "BA")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
federal <- read_csv2("D:/Covid/Covid-19_Bahia/federal_w_codes.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
centralidade <- read_csv2("D:/Covid/Covid-19_Bahia/new.dat.ba.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
clima <- read_csv2("D:/Covid/Covid-19_Bahia/climatic.br.csv") %>%
  mutate(precTotal = rowSums(.[3:7]),
         tmean = apply(.[8:17], 1, mean))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
meso <- read.csv2('D:/Covid/Covid-19_Bahia/meso.csv')
colnames(meso)[1] <- 'mesoregiao'
aero <- read.csv2('D:/Covid/Covid-19_Bahia/main.air.ba.csv')
```

Juntando as tabelas:

```
munis <- left_join(ibge, centralidade, by = c("cod_ibge" = "ibgecode")) %>%
  left_join(federal, by = c("cod_ibge" = "ibge")) %>%
  left_join(mun_covid, by = c("cod_ibge" = "city_ibge_code"))  %>%
  left_join(clima, by = c("cod_ibge" = "ibge")) %>%
  left_join(meso, by = c("cod_ibge" = "code")) %>%
  left_join(aero, by = c("cod_ibge" = "ibge")) %>%

  mutate(dens.road = ifelse(is.na(dens.road), 0, dens.road),
         afetados = ifelse(is.na(afetados), 0, afetados),
         airport = ifelse(is.na(airport), "NO", airport),
         confirmed = ifelse(is.na(confirmed), 0, confirmed),
         confirmed_per_100k_inhabitants = ifelse(is.na(confirmed_per_100k_inhabitan
ts),
                                                 0,
                                                 confirmed_per_100k_inhabitants))
munis <- data.frame(munis,
                    tam_pop_urb = munis$total.pop * (1 - munis$perc.rural))
```

Modelo logístico correlacionando as variáveis com presença ou ausência do vírus. As variáveis foram escolhidas visando diminuir a correlação entre elas e manter a expectativa teorica.

```
reg_log <- glm(afetados ~
                 airport + log(total.pop) + perc.rural +
                 log(eingen.cen.dist) + school.year +
                 perc.with.wages + dist.min +
                 tmean + log(precTotal),
               data = munis,
               family = binomial)
```

Checar a inflação do modelo:

```
# VIF
vif(reg_log)
```

```
##              airport       log(total.pop)           perc.rural
##             1.253375             1.603076             1.633194
## log(eingen.cen.dist)          school.year      perc.with.wages
##             1.397491             2.057346             1.155430
##             dist.min                tmean       log(precTotal)
##             1.547884             1.159342             1.515420
```

Resultado do modelo:

```
resu <- summary(reg_log)
```

Adicionado por Anderson (20/04/20)

```
cor(munis$perc.rural, munis$total.pop)
```

```
## [1] -0.2140484
```

```
cor(munis$tam_pop_urb, munis$total.pop)
```

```
## [1] 0.9987153
```

```
cor(munis$tam_pop_urb, munis$perc.rural)
```

```
## [1] -0.2287015
```

Modelo logístico com novas variáveis:

```
reg_log2 <- glm(afetados ~
            airport + log(total.pop)+ perc.rural +
              log(eingen.cen.dist) + school.year +
              perc.with.wages + dist.min.ilh.ssa +
              log(precTotal) + tmean,
            data = munis,
            family = binomial)
```

Checar a inflação do modelo:

```
# VIF
vif(reg_log2)
```

```
##              airport       log(total.pop)           perc.rural
##             1.201731             1.763695             1.723896
## log(eingen.cen.dist)          school.year      perc.with.wages
##             1.583172             2.078664             1.125361
##     dist.min.ilh.ssa       log(precTotal)                tmean
##             1.330776             1.211272             1.195101
```

Resultado do modelo:

```
resu2 <- summary(reg_log2)
```

Modelo logístico com novas variáveis: incluindo renda mensal (month.wages) e removendo variáveis para diminuir o VIF

```
reg_log3 <- glm(afetados ~
            airport + dist.min.ilh.ssa +
            log(eingen.cen.dist) +
            perc.with.wages  + month.wages +
            log(precTotal) + tmean,
            data = munis,
            family = binomial)
```

Checar a inflação do modelo:

```
# VIF
vif(reg_log3)
```

```
##             airport     dist.min.ilh.ssa log(eingen.cen.dist)
##            1.254318             1.166046             1.314663
##      perc.with.wages          month.wages       log(precTotal)
##            1.074428             1.652954             1.328441
##               tmean
##            1.131358
```

```
resu3 <- summary(reg_log3)
```

Modelo logístico apenas com mesorregiões:

```
reg_log4 <- glm(afetados ~
            mesoregiao,
            data = munis,
            family = binomial)
```

Na verdade, um qui-quadrado gourmet:

```
resu4 <- summary(reg_log4)
```

Modelo logístico apenas com estrutura da rede de transporte:

```
sim.roles<-ifelse(munis$roles=="network hub", "hub", munis$roles)

reg_log5 <- glm(afetados ~
            log(eingen.cen.dist)+sim.roles+as.factor(module),
            data = munis,
            family = binomial)
```

Na verdade, um qui-quadrado gourmet:

```
resu5 <- summary(reg_log5)
```

Comparando os modelos com AIC

```
aic.log <- AIC(reg_log,reg_log2,reg_log3, reg_log4, reg_log5)
```

```
## Warning in AIC.default(reg_log, reg_log2, reg_log3, reg_log4, reg_log5): models
## are not all fitted to the same number of observations
```

```
aic.log
```

	df <dbl>	AIC <dbl>
reg_log	10	361.2623
reg_log2	10	355.5557
reg_log3	8	375.1783
reg_log4	7	437.0670
reg_log5	9	420.3623

5 rows

Recalcular os valores de p por monte carlo, aleatorizando a variavel “afetados”.

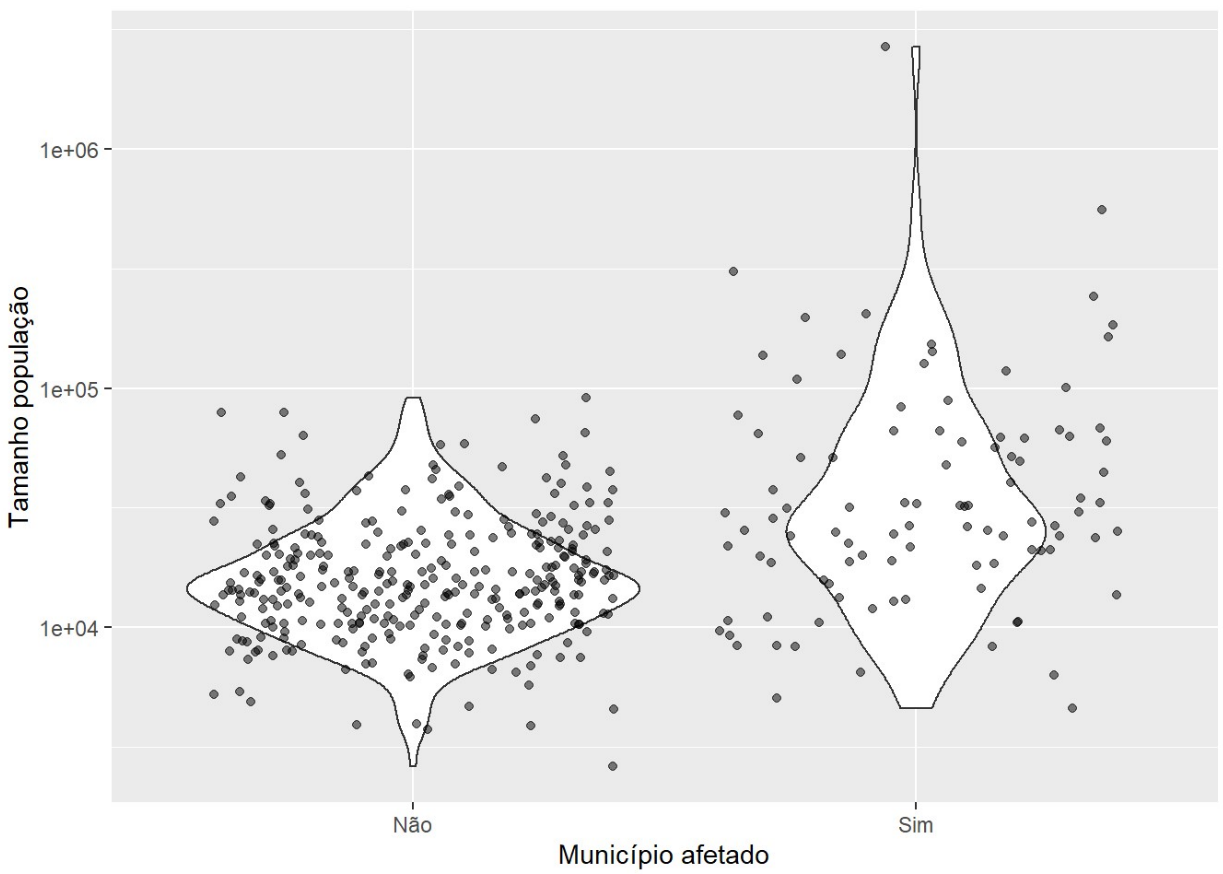
```
rept <- 1000
obs_z <- summary(reg_log)$coefficients[, 3]
obs <- coefficients(reg_log)
zs <- coefs <- matrix(ncol = length(obs), nrow = rept)
colnames(zs) <- colnames(coefs) <- names(obs)
for (i in 1:rept) {
  munis$rnd_afetados <- sample(munis$afetados)
  reg_log <- glm(
    rnd_afetados ~
      airport + log(total.pop) + perc.rural +
      log(eingen.cen.dist) + school.year +
      perc.with.wages + dist.min +
      tmean + precTotal,
    data = munis,
    family = binomial
  )
  zs[i, ] <- summary(reg_log)$coefficients[, 3]
  coefs[i, ] <- coefficients(reg_log)
}

for (j in 1:length(obs)) {
  maior <- (sum(obs[j] >= coefs[, j]) + 1 ) / (rept + 1) * 2
  menor <- (sum(obs[j] <= coefs[, j]) + 1) / (rept + 1) * 2
  resu$coefficients[j, 4] <- ifelse(maior > menor, menor, maior)
}
resu
```

```
##
## Call:
## glm(formula = afetados ~ airport + log(total.pop) + perc.rural +
##     log(eingen.cen.dist) + school.year + perc.with.wages + dist.min +
##     tmean + log(precTotal), family = binomial, data = munis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8220  -0.6400  -0.4323  -0.2020   2.7895
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -20.397086   5.943878  -3.432    0.002 **
## airportYES           -0.467196   0.745468  -0.627    0.627
## log(total.pop)        1.070007   0.246464   4.341    0.002 **
## perc.rural           -3.896961   0.939497  -4.148    0.002 **
## log(eingen.cen.dist)  0.028010   0.103659   0.270    0.829
## school.year          -0.135168   0.286022  -0.473    0.603
## perc.with.wages       2.206235   3.207786   0.688    0.390
## dist.min              0.002935   0.003468   0.846    0.348
## tmean                 0.060726   0.125298   0.485    0.583
## log(precTotal)        1.330068   0.568123   2.341    0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 450.48  on 404  degrees of freedom
## Residual deviance: 341.26  on 395  degrees of freedom
##   (12 observations deleted due to missingness)
## AIC: 361.26
##
## Number of Fisher Scoring iterations: 5
```

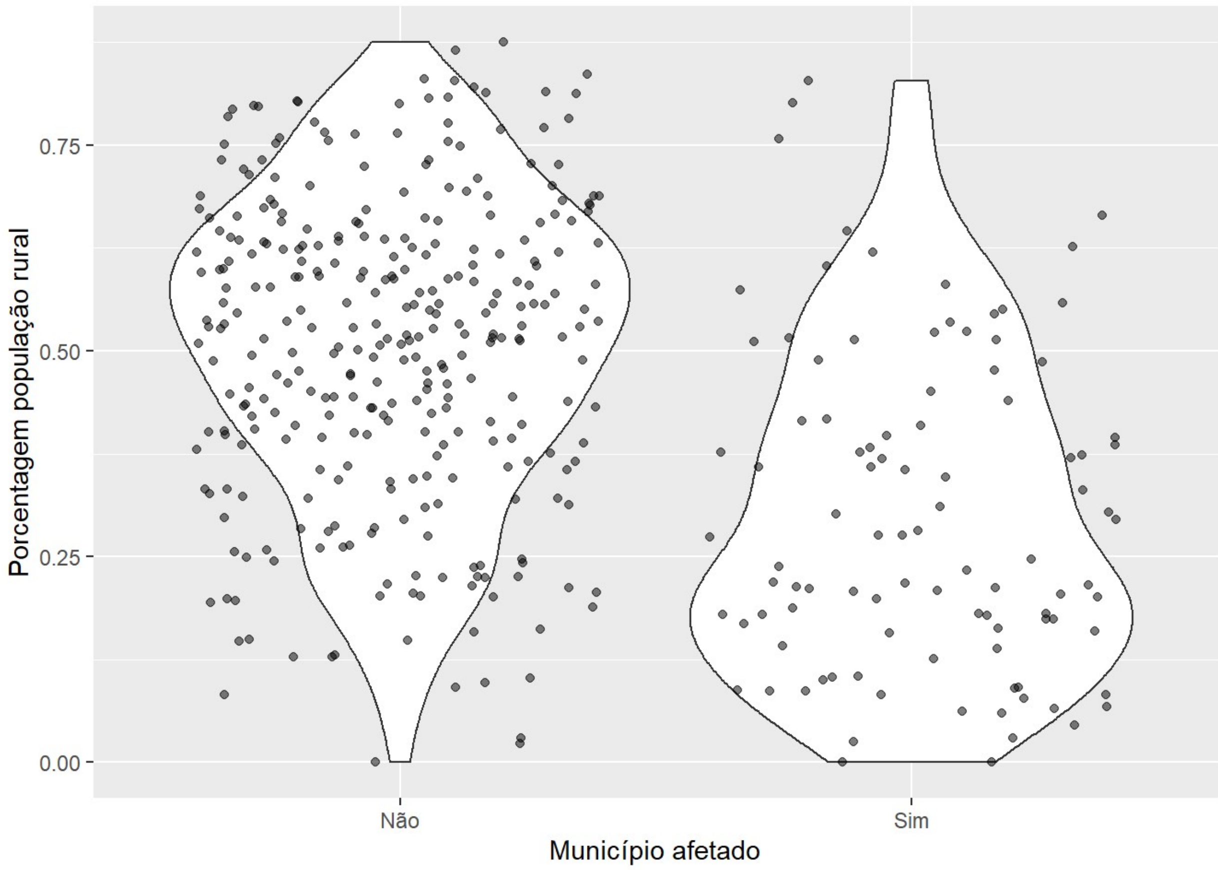
As simulações de monte carlo confirmam os resultados anteriores.

CONCLUSÃO: O modelo indica que municípios com uma população grande e urbana, têm maior probabilidade de serem afetados. Bem como municipios com mais chuva durante os últimos 4 meses.

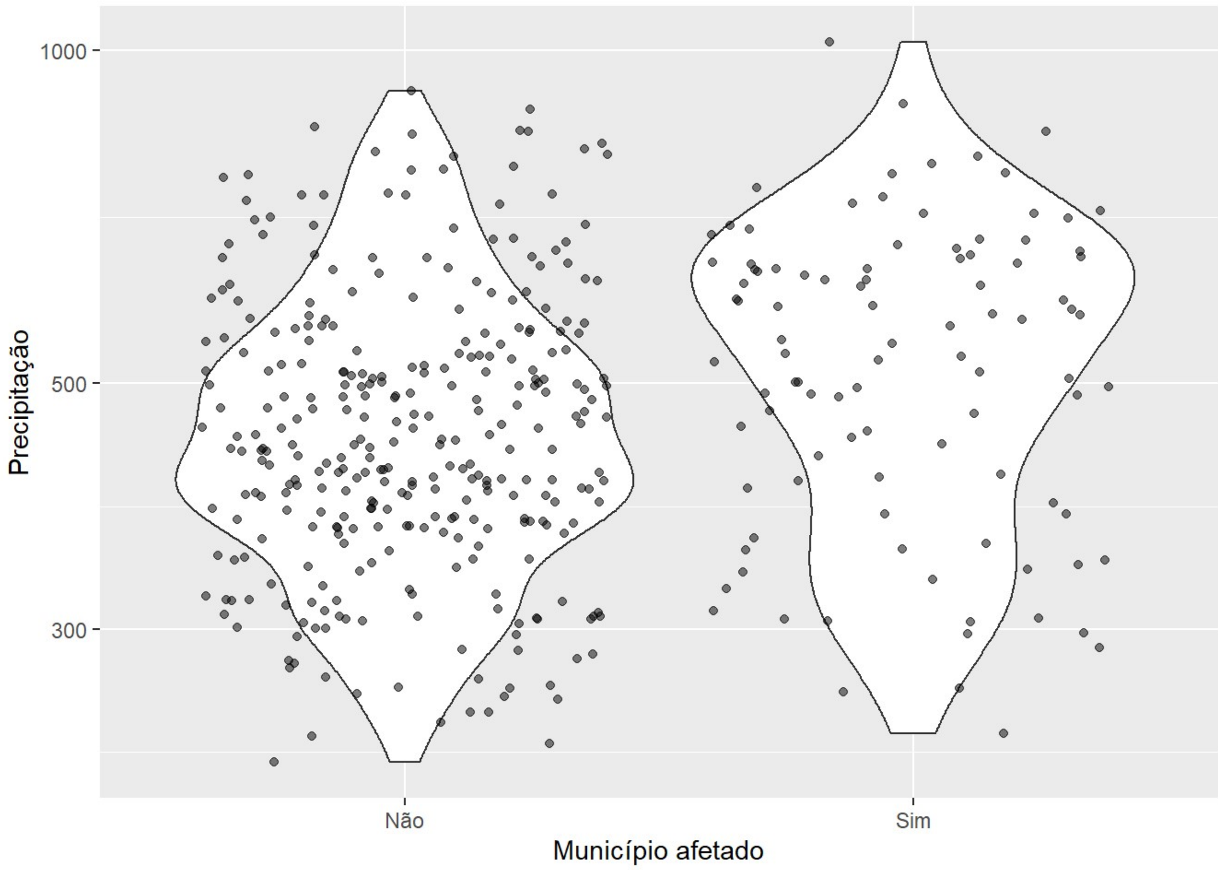
```
munis %>% mutate(afetados = ifelse(afetados == 1, "Sim", "Não")) %>%
ggplot(aes(x = afetados ,
           y = total.pop)) +
  geom_violin() +
  geom_jitter(alpha = .5) +
  scale_y_log10() +
  xlab("Município afetado") +
  ylab("Tamanho população")
```




```
munis %>% mutate(afetados = ifelse(afetados == 1, "Sim", "Não")) %>%
ggplot(aes(x = afetados ,
           y = perc.rural)) +
  geom_violin() +
  geom_jitter(alpha = .5) +
  xlab("Município afetado") +
  ylab("Porcentagem população rural")
```




```
munis %>% mutate(afetados = ifelse(afetados == 1, "Sim", "Não")) %>%
ggplot(aes(x = afetados,
           y = precTotal)) +
  geom_violin() +
  geom_jitter(alpha = .5) +
  scale_y_log10() +
  xlab("Município afetado") +
  ylab("Precipitação")
```



Correlacao com numero de casos

Principais fatores contribuindo para a quantidade de casos, nas cidades afetadas.

```
reg_N <- lm(log(confirmed_per_100k_inhabitants+1) ~
            airport + log(total.pop) + perc.rural +
            log(eingen.cen.dist) + school.year +
            perc.with.wages + dist.min +
              precTotal + tmean,
            data = filter(munis, confirmed > 0))
```

```
vif(reg_N)
```

```
##              airport       log(total.pop)           perc.rural
##             1.569572             3.288392             2.382952
## log(eingen.cen.dist)          school.year      perc.with.wages
##             2.103800             3.264889             1.380772
##             dist.min            precTotal                tmean
##             1.968708             1.772121             1.217400
```

```
resu <- summary(reg_N)
resu
```

```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ airport +
##     log(total.pop) + perc.rural + log(eingen.cen.dist) + school.year +
##     perc.with.wages + dist.min + precTotal + tmean, data = filter(munis,
##     confirmed > 0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0359 -0.5061 -0.0718  0.4793  2.1295
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.6074429  2.6561455   2.111   0.0376 *
## airportYES           0.1052273  0.3528710   0.298   0.7662
## log(total.pop)      -0.3129236  0.1383784  -2.261   0.0262 *
## perc.rural          -0.3826788  0.6459568  -0.592   0.5551
## log(eingen.cen.dist) -0.0132915  0.0781689  -0.170   0.8654
## school.year          0.0965923  0.1621021   0.596   0.5528
## perc.with.wages      0.4526233  2.3258227   0.195   0.8461
## dist.min            -0.0011584  0.0022661  -0.511   0.6105
## precTotal            0.0011080  0.0006753   1.641   0.1044
## tmean               -0.0634562  0.0754363  -0.841   0.4025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8057 on 89 degrees of freedom
## Multiple R-squared:  0.1882, Adjusted R-squared:  0.1061
## F-statistic: 2.293 on 9 and 89 DF,  p-value: 0.02299
```

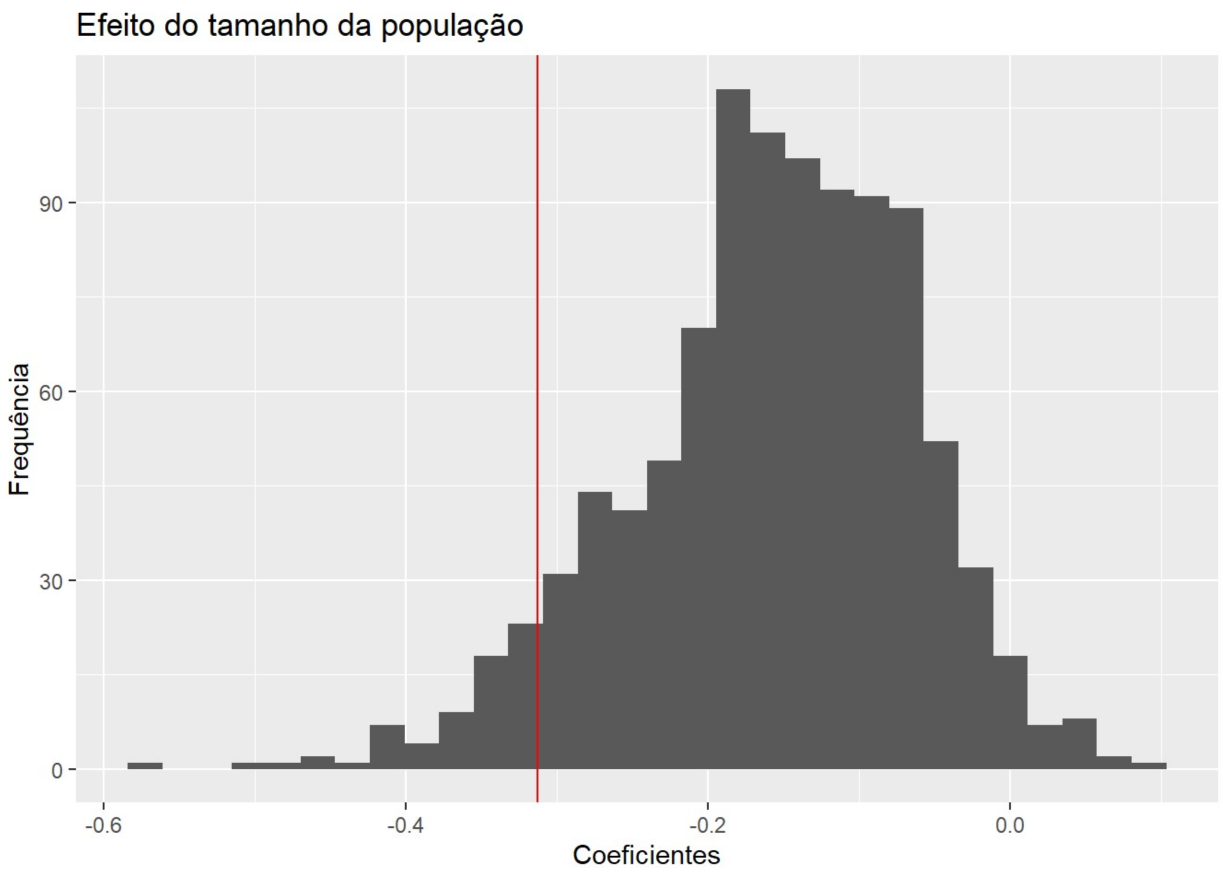
O resultado preliminar aponta para um efeito negativo da população sobre o número de casos a cada 100 mil habitantes, porém pode ser apenas um efeito matemático. Para avaliar isso, recalculamos abaixo os valore de p usando uma simulação da Monte Carlo, onde o numero de casos foi aleatorizado e foi recalculada o número de casos por 100k habitantes.

```
rept <- 1000
obs <- coef(reg_N)
coefs <- matrix(ncol = length(obs), nrow = rept)
colnames(coefs) <- names(obs)
for (i in 1:rept) {
 munis$rnd_cases <- (sample(munis$confirmed) / munis$total.pop) * 1000
  while(sum(munis$airport[munis$rnd_cases > 0] == "YES") < 2) {
    munis$rnd_cases <- (sample(munis$confirmed) / munis$total.pop) * 1000
  }
  reg_N <- lm(log(rnd_cases+1) ~
              airport + log(total.pop) + perc.rural +
              log(eingen.cen.dist) + school.year +
              perc.with.wages + dist.min +
              precTotal + tmean,
              data = filter(munis, rnd_cases > 0))
  coefs[i, ] <- coef(reg_N)
}

for (j in 1:length(obs)) {
  maior <- (sum(obs[j] >= coefs[, j]) + 1 ) / (rept + 1) * 2
  menor <- (sum(obs[j] <= coefs[, j]) + 1) / (rept + 1) * 2
  resu$coefficients[j, 4] <- ifelse(maior > menor, menor, maior)
}
resu
```

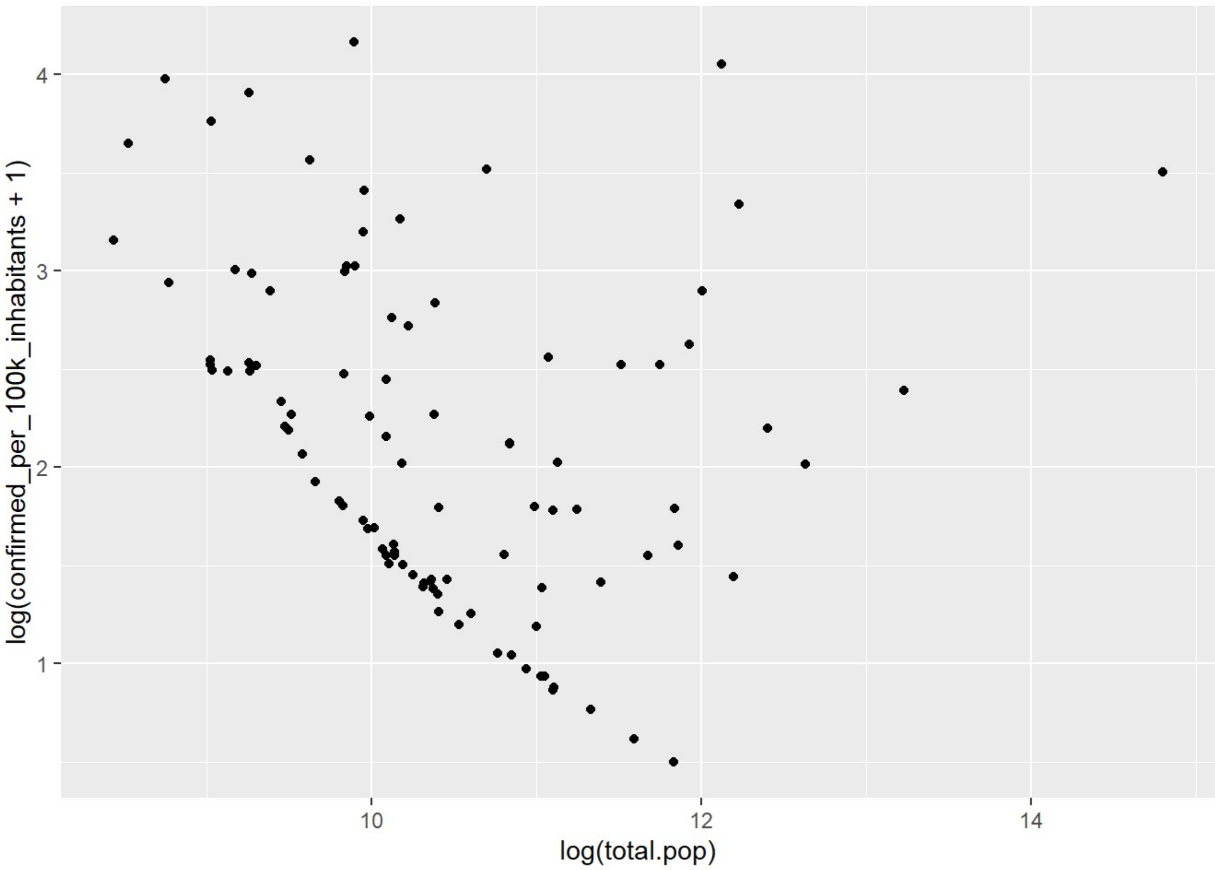
```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ airport +
##     log(total.pop) + perc.rural + log(eingen.cen.dist) + school.year +
##     perc.with.wages + dist.min + precTotal + tmean, data = filter(munis,
##     confirmed > 0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0359 -0.5061 -0.0718  0.4793  2.1295
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.6074429  2.6561455   2.111    0.042 *
## airportYES            0.1052273  0.3528710   0.298    0.653
## log(total.pop)       -0.3129236  0.1383784  -2.261    0.130
## perc.rural           -0.3826788  0.6459568  -0.592    0.298
## log(eingen.cen.dist) -0.0132915  0.0781689  -0.170    0.715
## school.year           0.0965923  0.1621021   0.596    0.567
## perc.with.wages       0.4526233  2.3258227   0.195    0.739
## dist.min             -0.0011584  0.0022661  -0.511    0.382
## precTotal             0.0011080  0.0006753   1.641    0.034 *
## tmean                -0.0634562  0.0754363  -0.841    0.198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8057 on 89 degrees of freedom
## Multiple R-squared:  0.1882, Adjusted R-squared:  0.1061
## F-statistic: 2.293 on 9 and 89 DF,  p-value: 0.02299
```

Mesmo calculando o p-valor usando simulação, o efeito da população se mantém. Como pode ser visto abaixo (em cinza, coeficientes simulados, em vermelho o valor observado para o tamanho da população).

```
ggplot(tibble(coefs = coefs[, 3]), aes(x = coefs)) +
  geom_histogram(bins = 30) +
  geom_vline(aes(xintercept = obs[3]), color = "red") +
  ggtitle("Efeito do tamanho da população") +
  ylab("Frequência") +
  xlab("Coeficientes")
```




```
munis %>% filter(confirmed > 0) %>%
ggplot(aes(y = log(confirmed_per_100k_inhabitants+1),
                  x = log(total.pop))) +
  geom_point()
```



Correlacao com numero de casos (Novas variáveis)

Principais fatores contribuindo para a quantidade de casos, nas cidades afetadas.

```
reg_N2 <- lm(log(confirmed_per_100k_inhabitants + 1) ~
            airport + perc.rural +
              log(eingen.cen.dist) + school.year +
              perc.with.wages + dist.min +
              precTotal + tmean + log(tam_pop_urb) +
              mesoregiao + dist.min.ilh.ssa,
            data = filter(munis, confirmed > 0))
```

```
vif(reg_N2)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## airport                 1.692637  1        1.301014
## perc.rural              3.633065  1        1.906060
## log(eingen.cen.dist)    2.634214  1        1.623026
## school.year             4.726109  1        2.173961
## perc.with.wages         1.864365  1        1.365418
## dist.min                3.410410  1        1.846729
## precTotal               4.917025  1        2.217437
## tmean                   3.374014  1        1.836849
## log(tam_pop_urb)        5.276422  1        2.297046
## mesoregiao            175.858395  6        1.538498
## dist.min.ilh.ssa        4.870752  1        2.206978
```

```
resu2 <- summary(reg_N2)
resu2
```

```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ airport +
##     perc.rural + log(eingen.cen.dist) + school.year + perc.with.wages +
##     dist.min + precTotal + tmean + log(tam_pop_urb) + mesoregiao +
##     dist.min.ilh.ssa, data = filter(munis, confirmed > 0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20631 -0.38892 -0.00866  0.34907  1.61172
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             1.6497318  2.5884189   0.637 0.525672
## airportYES                              0.1498504  0.2810627   0.533 0.595367
## perc.rural                              1.0899370  0.6117562   1.782 0.078508
## log(eingen.cen.dist)                   -0.1107033  0.0670893  -1.650 0.102750
## school.year                             0.6220506  0.1495899   4.158 7.84e-05
## perc.with.wages                        -1.2679440  2.0728935  -0.612 0.542443
## dist.min                                0.0078700  0.0022877   3.440 0.000917
## precTotal                               0.0002883  0.0008628   0.334 0.739121
## tmean                                  -0.1245617  0.0963237  -1.293 0.199588
## log(tam_pop_urb)                       -0.1355991  0.1149163  -1.180 0.241420
## mesoregiaoCentro-Sul Baiano             0.5998163  0.3090413   1.941 0.055707
## mesoregiaoExtremo Oeste Baiano          0.7037073  0.7256316   0.970 0.335005
## mesoregiaoMetropolitana de Salvador    -0.3674579  0.3644122  -1.008 0.316249
## mesoregiaoNordeste Baiano              -0.5838533  0.2949752  -1.979 0.051134
## mesoregiaoSul Baiano                    1.0189581  0.3587113   2.841 0.005678
## mesoregiaoVale São-Franciscano da Bahia  0.7477675  0.4313301   1.734 0.086742
## dist.min.ilh.ssa                       -0.0036743  0.0008461  -4.343 4.00e-05
##
## (Intercept)
## airportYES
## perc.rural                              .
## log(eingen.cen.dist)
## school.year                            ***
## perc.with.wages
## dist.min                               ***
## precTotal
## tmean
## log(tam_pop_urb)
## mesoregiaoCentro-Sul Baiano             .
## mesoregiaoExtremo Oeste Baiano
## mesoregiaoMetropolitana de Salvador
## mesoregiaoNordeste Baiano               .
## mesoregiaoSul Baiano                   **
## mesoregiaoVale São-Franciscano da Bahia .
## dist.min.ilh.ssa                       ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6179 on 82 degrees of freedom
```

```
## Multiple R-squared:   0.56,  Adjusted R-squared:  0.4742
## F-statistic: 6.523 on 16 and 82 DF,  p-value: 3.509e-09
```

Reajustando o modelo 2 para diminuir o vif ou variáveis redundantes (removendo popurbana, distmin,meso e escolaridade)

```
reg_N3 <- lm(log(confirmed_per_100k_inhabitants + 1) ~
            airport + perc.rural +
              log(eingen.cen.dist) +
              perc.with.wages  +
              precTotal + tmean +
               dist.min.ilh.ssa,
            data = filter(munis, confirmed > 0))
```

```
vif(reg_N3)
```

```
##              airport           perc.rural log(eingen.cen.dist)
##             1.188778             1.921034             1.406985
##      perc.with.wages            precTotal                tmean
##             1.284405             1.419788             1.109832
##     dist.min.ilh.ssa
##             1.149634
```

```
resu3 <- summary(reg_N3)
resu3
```

```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ airport +
##     perc.rural + log(eingen.cen.dist) + perc.with.wages + precTotal +
##     tmean + dist.min.ilh.ssa, data = filter(munis, confirmed >
##     0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5151 -0.4784 -0.0848  0.5041  1.7202
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.7336430  2.1978664   2.154  0.03390 *
## airportYES          -0.0149474  0.2761452  -0.054  0.95695
## perc.rural           0.3599132  0.5215251   0.690  0.49188
## log(eingen.cen.dist) -0.1063033  0.0574829  -1.849  0.06766 .
## perc.with.wages      1.2454936  2.0171049   0.617  0.53847
## precTotal            0.0017126  0.0005435   3.151  0.00220 **
## tmean               -0.1719498  0.0647670  -2.655  0.00936 **
## dist.min.ilh.ssa    -0.0024436  0.0004819  -5.071 2.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7245 on 91 degrees of freedom
## Multiple R-squared:  0.3289, Adjusted R-squared:  0.2772
## F-statistic:  6.37 on 7 and 91 DF,  p-value: 4.213e-06
```

Modelo somente mesorregiões

```
reg_N4 <- lm(log(confirmed_per_100k_inhabitants + 1) ~
            mesoregiao,
            data = filter(munis, confirmed > 0))
```

Resultado

```
resu4 <- summary(reg_N4)
resu4
```

```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ mesoregiao,
##     data = filter(munis, confirmed > 0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74707 -0.42955 -0.02197  0.36171  1.68689
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                1.74494    0.21411   8.150 1.76e-12
## mesoregiaoCentro-Sul Baiano                0.64638    0.26904   2.403 0.018290
## mesoregiaoExtremo Oeste Baiano            -0.76437    0.46253  -1.653 0.101824
## mesoregiaoMetropolitana de Salvador        0.07292    0.29092   0.251 0.802634
## mesoregiaoNordeste Baiano                 -0.22430    0.29642  -0.757 0.451175
## mesoregiaoSul Baiano                       0.93851    0.24632   3.810 0.000251
## mesoregiaoVale São-Franciscano da Bahia -0.11598    0.34334  -0.338 0.736291
##
## (Intercept)                               ***
## mesoregiaoCentro-Sul Baiano               *
## mesoregiaoExtremo Oeste Baiano
## mesoregiaoMetropolitana de Salvador
## mesoregiaoNordeste Baiano
## mesoregiaoSul Baiano                      ***
## mesoregiaoVale São-Franciscano da Bahia
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7101 on 92 degrees of freedom
## Multiple R-squared:  0.3481, Adjusted R-squared:  0.3056
## F-statistic: 8.187 on 6 and 92 DF,  p-value: 4.198e-07
```

Modelo somente estrutura da rede de transporte

```
reg_N5 <- lm(log(confirmed_per_100k_inhabitants + 1) ~
            as.factor(module)+log(eingen.cen.dist)+roles,
            data = filter(munis, confirmed > 0))
```

```
vif(reg_N5)
```

```
##                          GVIF Df GVIF^(1/(2*Df))
## as.factor(module)    1.516834  5        1.042543
## log(eingen.cen.dist) 1.418458  1        1.190991
## roles                1.792460  3        1.102152
```

Resultado

```
resu5 <- summary(reg_N5)
resu5
```

```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ as.factor(module) +
##     log(eingen.cen.dist) + roles, data = filter(munis, confirmed >
##     0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65216 -0.46614  0.03754  0.46433  1.86725
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.238776   0.297048   4.170 7.05e-05 ***
## as.factor(module)2    0.337400   0.296513   1.138  0.25822
## as.factor(module)3    0.049528   0.380407   0.130  0.89670
## as.factor(module)4    0.284319   0.256922   1.107  0.27143
## as.factor(module)5    1.025204   0.233087   4.398 3.01e-05 ***
## as.factor(module)6   -0.119228   0.258372  -0.461  0.64560
## log(eingen.cen.dist) -0.117928   0.058105  -2.030  0.04539 *
## roleshub              0.462800   0.431674   1.072  0.28657
## rolesnetwork hub      1.625358   0.585813   2.775  0.00674 **
## rolesperipheral       0.002341   0.214615   0.011  0.99132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7293 on 89 degrees of freedom
## Multiple R-squared:  0.3348, Adjusted R-squared:  0.2675
## F-statistic: 4.976 on 9 and 89 DF,  p-value: 2.021e-05
```

Modelo incluindo renda mensal (month.wages)

```
reg_N6 <- lm(log(confirmed_per_100k_inhabitants + 1) ~
            airport + dist.min.ilh.ssa +
            log(eingen.cen.dist) +
            perc.with.wages  + month.wages +
            log(precTotal) + tmean,
            data = munis)
```

Checar a inflação do modelo:

```
# VIF
vif(reg_N6)
```

```
##              airport     dist.min.ilh.ssa log(eingen.cen.dist)
##             1.224698             1.205720             1.357233
##      perc.with.wages          month.wages       log(precTotal)
##             1.145097             1.793627             1.346965
##                tmean
##             1.104157
```

```
resu6 <- summary(reg_N6)
resu6
```

```
##
## Call:
## lm(formula = log(confirmed_per_100k_inhabitants + 1) ~ airport +
##     dist.min.ilh.ssa + log(eingen.cen.dist) + perc.with.wages +
##     month.wages + log(precTotal) + tmean, data = munis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8704 -0.5330 -0.2127  0.0695  3.1691
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -3.3611447  1.6259942  -2.067   0.0394 *
## airportYES           -0.0840687  0.2629317  -0.320   0.7493
## dist.min.ilh.ssa     -0.0012936  0.0002946  -4.391 1.45e-05 ***
## log(eingen.cen.dist)  0.0018553  0.0315266   0.059   0.9531
## perc.with.wages       0.2534496  1.0096857   0.251   0.8019
## month.wages           0.0022647  0.0004203   5.388 1.22e-07 ***
## log(precTotal)        0.4492899  0.1741777   2.579   0.0103 *
## tmean                 0.0043328  0.0387572   0.112   0.9110
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.903 on 397 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.219,  Adjusted R-squared:  0.2052
## F-statistic:  15.9 on 7 and 397 DF,  p-value: < 2.2e-16
```

AIC

```
AIC(reg_N,reg_N3,reg_N4, reg_N5, reg_N6)
```

```
## Warning in AIC.default(reg_N, reg_N3, reg_N4, reg_N5, reg_N6): models are not
## all fitted to the same number of observations
```

	df <dbl>	AIC <dbl>
reg_N	11	167.6453

	df <dbl>	AIC <dbl>
reg_N3	9	226.7859
reg_N4	8	221.9108
reg_N5	11	229.9133
reg_N6	9	1076.5971

5 rows

Correlacao com o tempo até o primeiro caso

```
covid.day <- get_corona_br(
  dir = "output",
  filename = "corona_brasil",
  cidade = NULL,
  uf = 'BA',
  ibge_cod = NULL,
  by_uf = FALSE
)
covid.day <- data.frame(covid.day,
                        afetados = ifelse(covid.day$confirmed > 0, 1, 0))
```

```
covid.day <- left_join(ibge, centralidade, by = c("cod_ibge" = "ibgecode")) %>%
  left_join(federal, by = c("cod_ibge" = "ibge")) %>%
  left_join(covid.day, by = c("cod_ibge" = "city_ibge_code"))  %>%
  left_join(clima, by = c("cod_ibge" = "ibge")) %>%
  left_join(meso, by = c("cod_ibge" = "code")) %>%

  mutate(dens.road = ifelse(is.na(dens.road), 0, dens.road),
         afetados = ifelse(is.na(afetados), 0, afetados),
         airport = ifelse(is.na(airport), "NO", airport),
         confirmed = ifelse(is.na(confirmed), 0, confirmed),
         confirmed_per_100k_inhabitants = ifelse(is.na(confirmed_per_100k_inhabitan
ts),
                                                 0,
                                                 confirmed_per_100k_inhabitants))
cod <- unique(covid.day$cod_ibge)
tempo <- data.frame(matrix(ncol = 2, nrow = length(cod)))
colnames(tempo) <- c('cod_ibge', 'tempo_1')
tempo[, 1] <- cod

for(i in 1:length(cod)){
  tab.cod <- data.frame(covid.day[covid.day$cod_ibge == cod[i], ])
  tab.cod <- tab.cod[order(tab.cod$date), ]
  if(sum(tab.cod$confirmed > 0) > 0){ # Mudar de acordo com a qtd de casos
  primeiro <- tab.cod[min(which(tab.cod$confirmed > 0)), 'date']
  baseline <- as.Date.character('2020-03-06')
  tempo [i, 2] <- as.numeric(difftime(primeiro, baseline, 'days'))
  }
  if(sum(tab.cod$confirmed > 0) == 0){
    tempo [i, 2] <- NA
  }
}

munis <- merge(munis, tempo, by = 'cod_ibge',
               all.x = T, all.y = F, sort = F)
```

Principais fatores contribuindo para a quantidade de casos Removi distmin porque é redundante com dis.min.ilh.ssa Removi meso por causa do vif alto

```
reg_T <- lm(tempo_1 ~
              airport +
              log(eingen.cen.dist) + school.year +
              perc.with.wages  +
              precTotal + tmean +
              dist.min.ilh.ssa,
            data = filter(munis, confirmed > 0))
```

```
vif(reg_T)
```

```
##             airport log(eingen.cen.dist)          school.year
##            1.231631             1.564045             1.871751
##     perc.with.wages            precTotal                tmean
##            1.230787             1.176542             1.168576
##    dist.min.ilh.ssa
##            1.083086
```

```
resu <- summary(reg_T)
resu
```

```
##
## Call:
## lm(formula = tempo_1 ~ airport + log(eingen.cen.dist) + school.year +
##     perc.with.wages + precTotal + tmean + dist.min.ilh.ssa, data = filter(munis,
##     confirmed > 0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4128  -6.5659  -0.7582   5.7461  20.6121
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           8.595e+01  2.530e+01   3.397 0.001012 **
## airportYES            1.327e-01  3.405e+00   0.039 0.968991
## log(eingen.cen.dist) -3.347e-01  7.343e-01  -0.456 0.649649
## school.year          -4.996e+00  1.337e+00  -3.737 0.000325 ***
## perc.with.wages      -4.808e+00  2.392e+01  -0.201 0.841163
## precTotal            -7.649e-05  5.995e-03  -0.013 0.989847
## tmean                -7.725e-01  8.052e-01  -0.959 0.339877
## dist.min.ilh.ssa      1.734e-03  5.667e-03   0.306 0.760393
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.777 on 91 degrees of freedom
## Multiple R-squared:  0.2792, Adjusted R-squared:  0.2237
## F-statistic: 5.035 on 7 and 91 DF,  p-value: 7.43e-05
```

Prever municipios com maior probabilidade de serem afetados

Para isso usamos um random forest.

```r
# data_mod2 <- data_mod %>% na.omit() %>%
#   mutate(total.pop = log(total.pop),
#          eingen.cen.dist = log(eingen.cen.dist),
#          afetados = as.factor(afetados),
#          airport = as.factor(airport))
#
# reg_log <- randomForest(afetados ~
#                  airport + total.pop + perc.rural +
#                  eingen.cen.dist + school.year +
#                  perc.with.wages + dist.min,
#                data = data_mod2,
#                importance = TRUE)
```

Fazer um mapa de vulnerabilidade aqui.

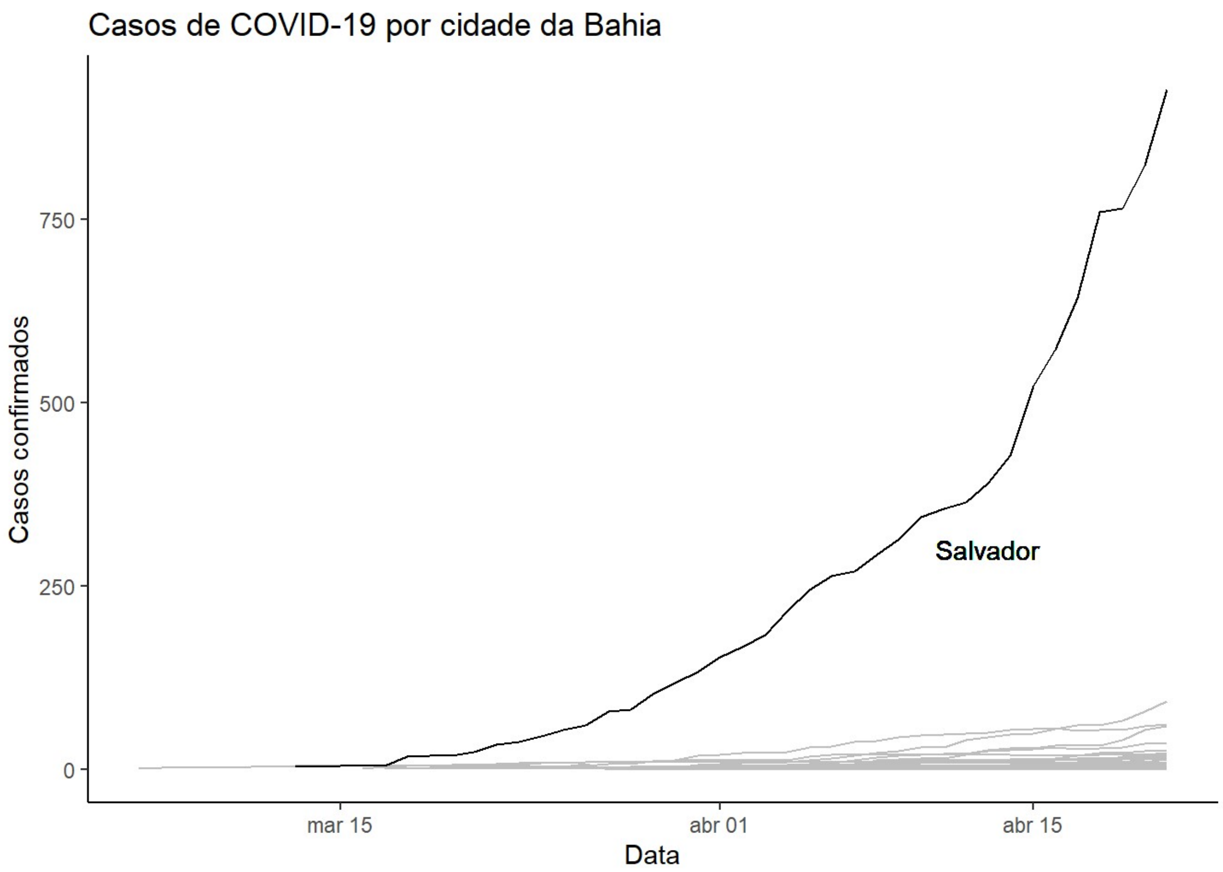
Taxa de crescimento dos casos de covid-19

Calcular a taxa

```r
# USAR MESMA ESTRATEGIA DO OUTRO ARTIGO
```

Gráfico de todos os municipios

```r
cols <- ifelse(levels(covid$city) == "Salvador", "black", "gray")
covid %>%
  ggplot(aes(x = date, y = confirmed, color = city)) +
  scale_color_manual(values = cols) +
  geom_line() +
  ggtitle("Casos de COVID-19 por cidade da Bahia") +
  ylab("Casos confirmados") +
  xlab("Data") +
  theme_classic() +
  theme(legend.position = "none") +
  geom_text(aes(x = date[1] - 8, y = 300,
                label = "Salvador"), color = "black")
```



Como a taxa tem variado ao longo do tempo?

```
covid <- as_tibble(get_corona_br(uf = "BA"))
```

Pequenos ajustes na tabela:

```
covid.ba <- covid %>%
  filter(place_type == "state")

covid.ci <- covid %>%
  filter(place_type == "city")
```

Segunda abordagem para crescimento exponencial

```
gm_mean <- function(x, na.rm = TRUE){
  exp(sum(log(x[x > 0]), na.rm=na.rm) / length(x))
}
r_calc <- function(x) {
  gm_mean(x[2:length(x)] / x[1:(length(x)-1)])
}
```

Taxa de crescimento Bahia

```r
casos.ba <- covid.ba$confirmed[nrow(covid.ba):1] #backwards
tempo.ba <- 1:length(casos.ba)
head(covid.ba)
```

date	state	city	place_type	confirmed	deat…	is_last	estimated_popul
<date>	<fctr>	<fctr>	<fctr>	<int>	<int>	<fctr>	
2020-04-21	BA		state	1504	48	True	
2020-04-20	BA		state	1377	47	False	
2020-04-19	BA		state	1249	45	False	
2020-04-18	BA		state	1200	40	False	
2020-04-17	BA		state	1064	36	False	
2020-04-16	BA		state	967	34	False	

6 rows | 1-8 of 11 columns

```r
exp.ba<- lm(log(casos.ba)~tempo.ba)
tax.ba<-coef(exp.ba)[2]

r.time.b<- data.frame(tempo=tempo.ba,
           confirmados=casos.ba,
           data=covid.ba$date[nrow(covid.ba):1],
           taxa=NA,
           r.squ=NA,
           taxa2=NA) #segunda abordagem

for (i in 5:length(casos.ba))
{
  exp.temp<-lm(log(casos.ba[1:i])~tempo.ba[1:i])
  r.time.b[i,4]<-coef(exp.temp)[2]
  r.time.b[i,5]<-summary(exp.temp)$r.squared
  r.time.b[i,6]<-r_calc(casos.ba[1:i])
}
```

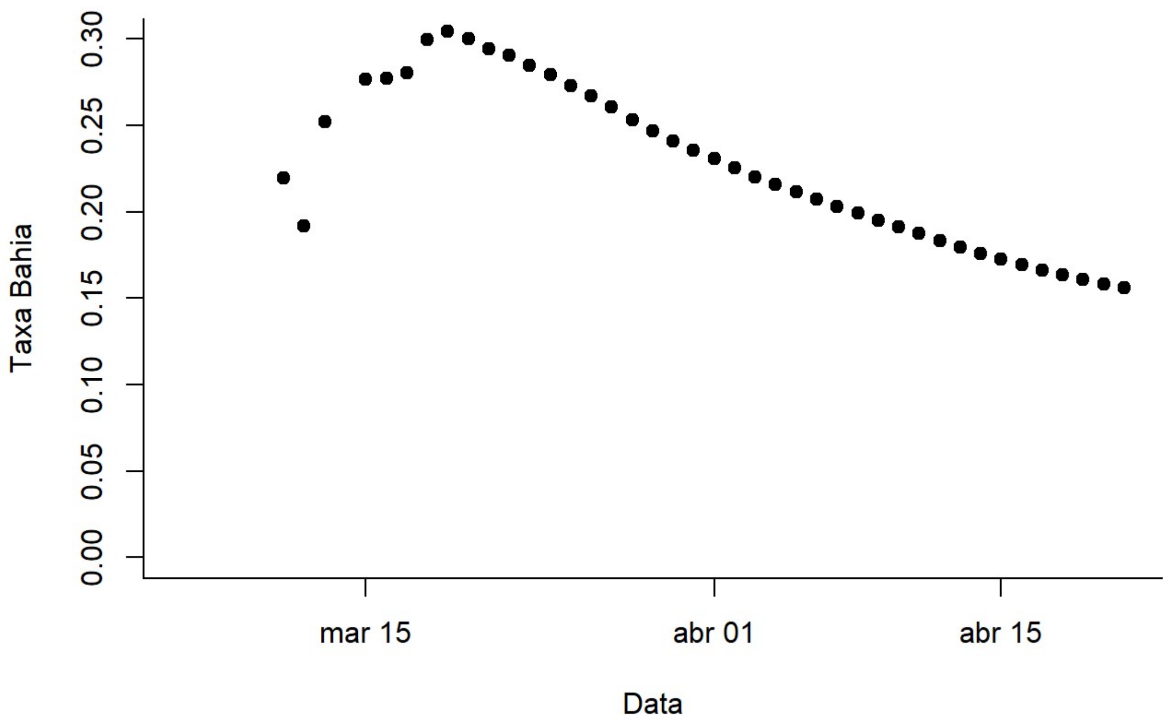
Correlação entre as duas abordagens

```r
cor.test(r.time.b[,4], r.time.b[,6])
```

```
##
##  Pearson's product-moment correlation
##
## data:  r.time.b[, 4] and r.time.b[, 6]
## t = 16.428, df = 39, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8802158 0.9649085
## sample estimates:
##       cor
## 0.9347399
```

Taxa de crescimento Bahia

```
plot(r.time.b$data, r.time.b$taxa,
     bty="l", pch=19, ylim=c(0,0.3),
     xlab="Data", ylab="Taxa Bahia")
```



Taxa de crescimento Municípios

```r
city.codes<-na.omit(unique(covid.ci$city_ibge_code))
city.n<-length(city.codes)
w=1
list.ci<-list()
for (w in 1:city.n)
{
  covid.temp <- covid.ci %>%
  filter(city_ibge_code == city.codes[w])
  n.temp<-nrow(covid.temp)
  casos.temp<-covid.temp$confirmed[n.temp:1]
  tempo.temp<-1:n.temp

  list.ci[[w]]<-data.frame(tempo=tempo.temp, #tempo
                      casos=casos.temp) #casos

}
names(list.ci)<-city.codes
n.obs.ci<-sapply(list.ci, nrow) #tamanho das séries temporais

#filtrando cidades com no mínimo dez dias com corona
new.list.ci<-list.ci[n.obs.ci>9]

#Ajustado modelo exponencial para cada munícipio
exp.ci<-log(tempo)~casos #equação
mod.exp.ci<-lapply(new.list.ci, lm, formula=exp.ci) #modelo exponencial
coe.exp.ci<-lapply(mod.exp.ci, coef) #coeficientes
r.ci<-sapply(coe.exp.ci, "[", 2) #só a inclinação ("taxa r")
sum.exp.ci<-lapply(mod.exp.ci, summary) #sumário dos modelos
r.squ.ci<-sapply(sum.exp.ci, "[", "r.squared")
r.squ.ci<-unlist(r.squ.ci)

#ajustando na segunda abordagem
r.ci2<-numeric()
for (i in 1:length(new.list.ci))
{
  r.ci2[i]<-r_calc(new.list.ci[[i]][,2])
}


# Montando a planilha

r.mun.dat<-data.frame(city_ibge_code=names(new.list.ci),
           taxa=r.ci,
           r.square=r.squ.ci,
           taxa2=r.ci2)
only80<-r.mun.dat %>%
  filter(r.square>0.8)
cor.test(only80[,2], only80[,4])
```

```
##
##  Pearson's product-moment correlation
##
## data:  only80[, 2] and only80[, 4]
## t = -2.759, df = 8, p-value = 0.02471
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9223738 -0.1224939
## sample estimates:
##        cor
## -0.6982661
```

Que fatores afetam a taxa de crescimento?

Quando ocorreram os picos?

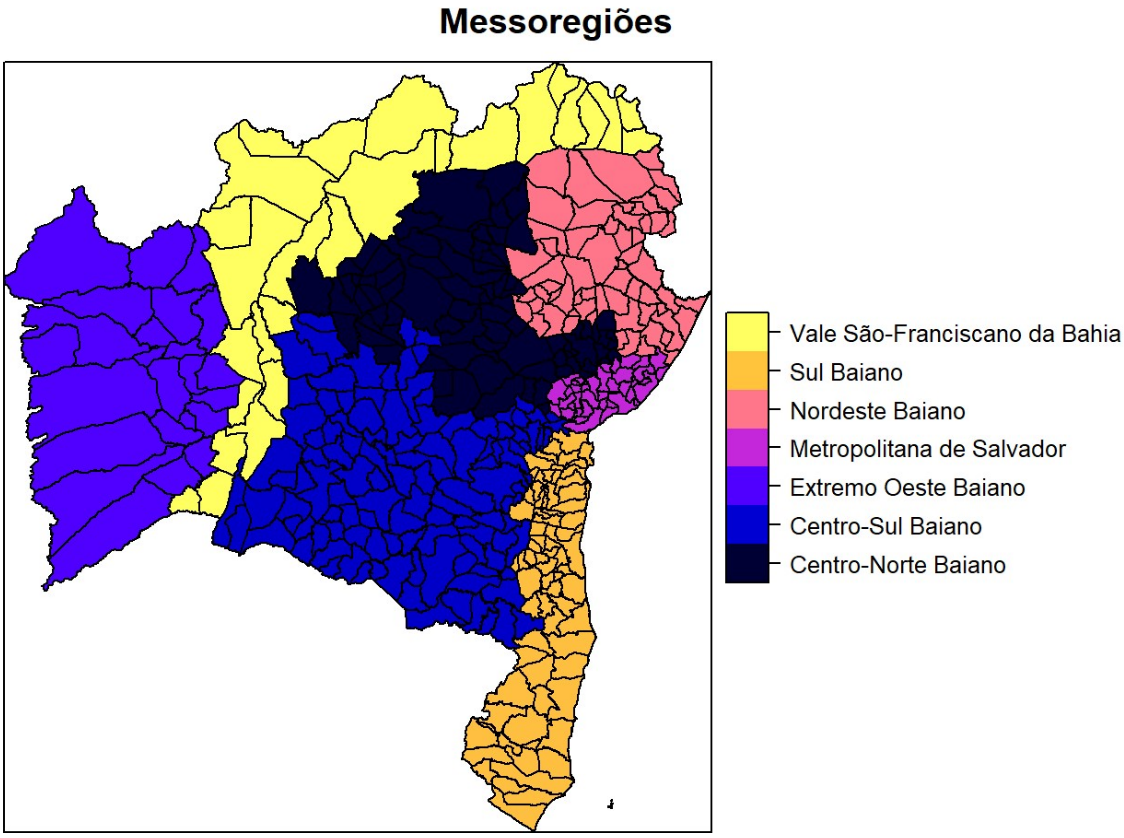
Qual a meta de quarentena para evitar colapso do sistema?

Mapas

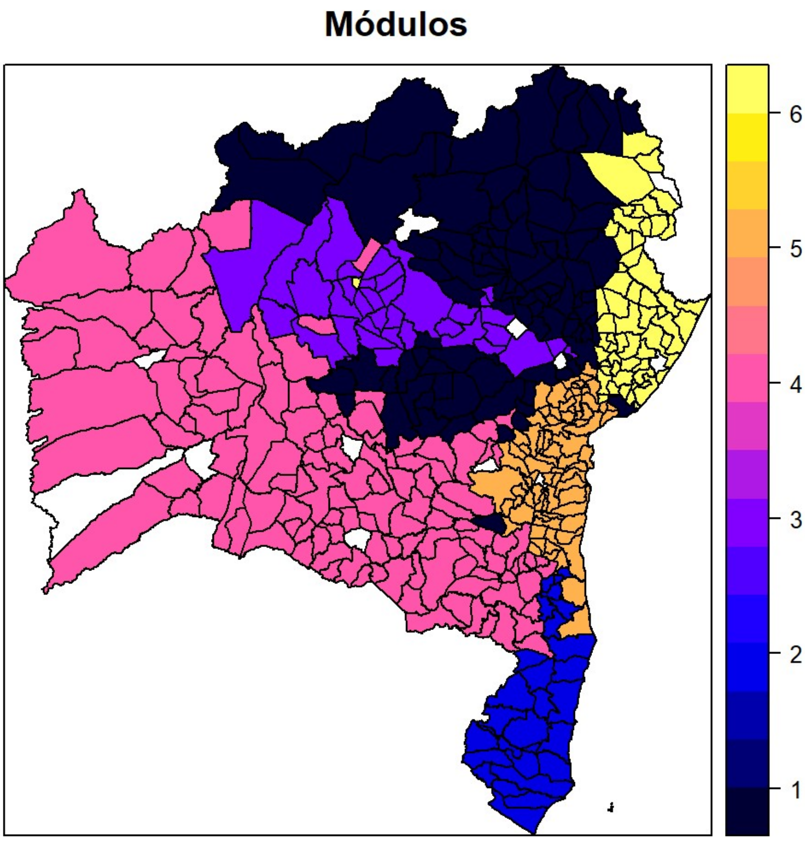
```
ba.shp<-readOGR("D:/Covid/Covid-19_Bahia/bahia.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "D:\Covid\Covid-19_Bahia\bahia.shp", layer: "bahia"
## with 417 features
## It has 2 fields
```

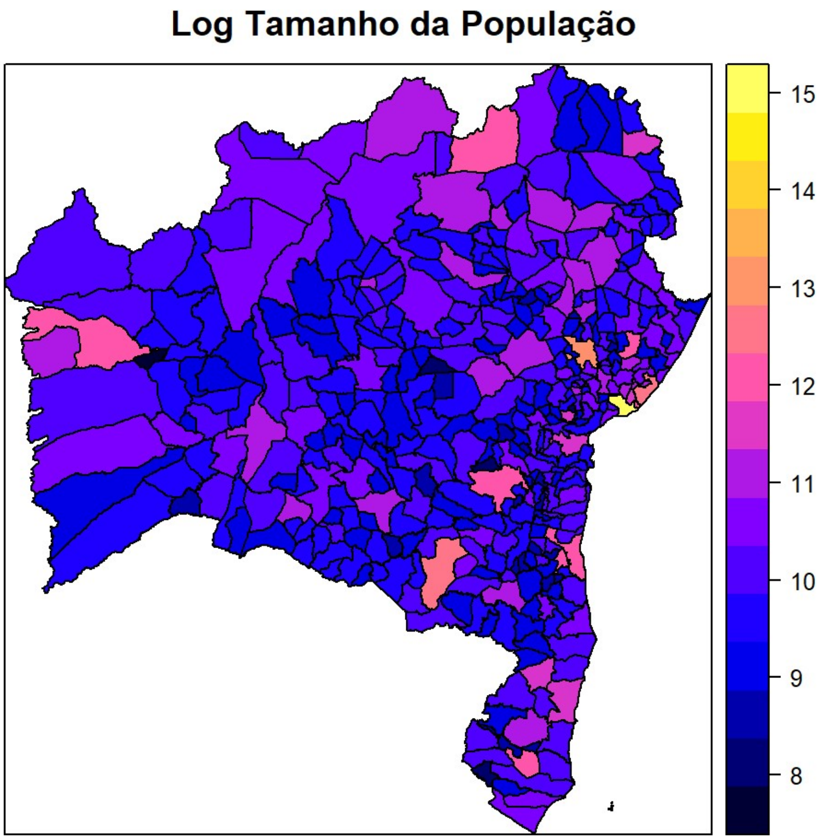
```
x.ibge<-match(ba.shp$CD_GEOCMU, munis$cod_ibge)
x.ibge2<-match(ba.shp$CD_GEOCMU, filter(munis, confirmed > 0)$cod_ibge)
ba.shp$mesoregiao<-munis$mesoregiao[x.ibge]
ba.shp$module<-munis$module[x.ibge]
ba.shp$total.pop<-log(munis$total.pop)[x.ibge]
ba.shp$perc.rural<-munis$perc.rural[x.ibge]*100
ba.shp$dist.min.ilh.ssa<-munis$dist.min.ilh.ssa[x.ibge]
ba.shp$precTotal<-log(munis$precTotal)[x.ibge]
ba.shp$afetados<-munis$afetados[x.ibge]
ba.shp$res_casos<-resid(reg_N3)[x.ibge2]
ba.shp$pred_afetados<-predict(reg_log2, type="response")[x.ibge]

sp::spplot(ba.shp, "mesoregiao", main="Messoregiões")
```



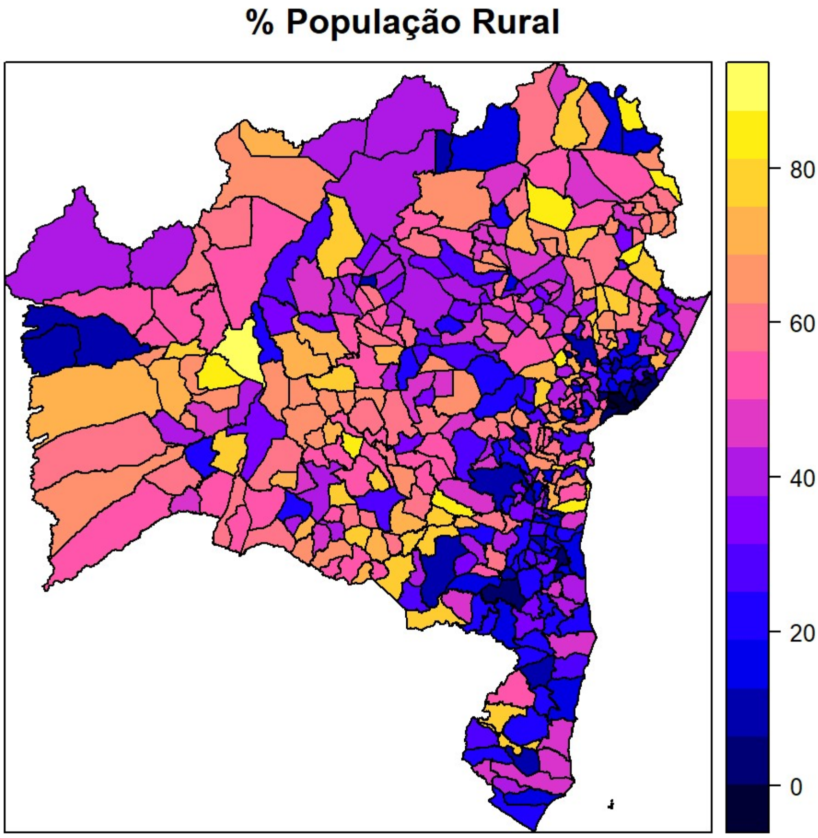

```
sp::spplot(ba.shp, "module", main="Módulos")
```



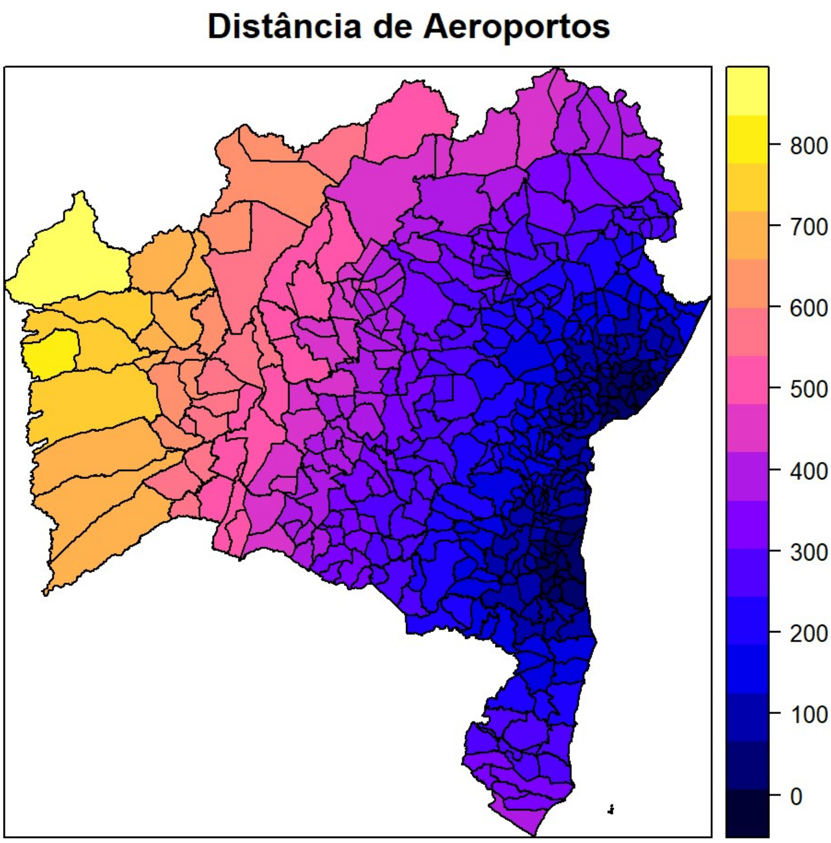
```
sp::spplot(ba.shp, "total.pop", main="Log Tamanho da População")
```



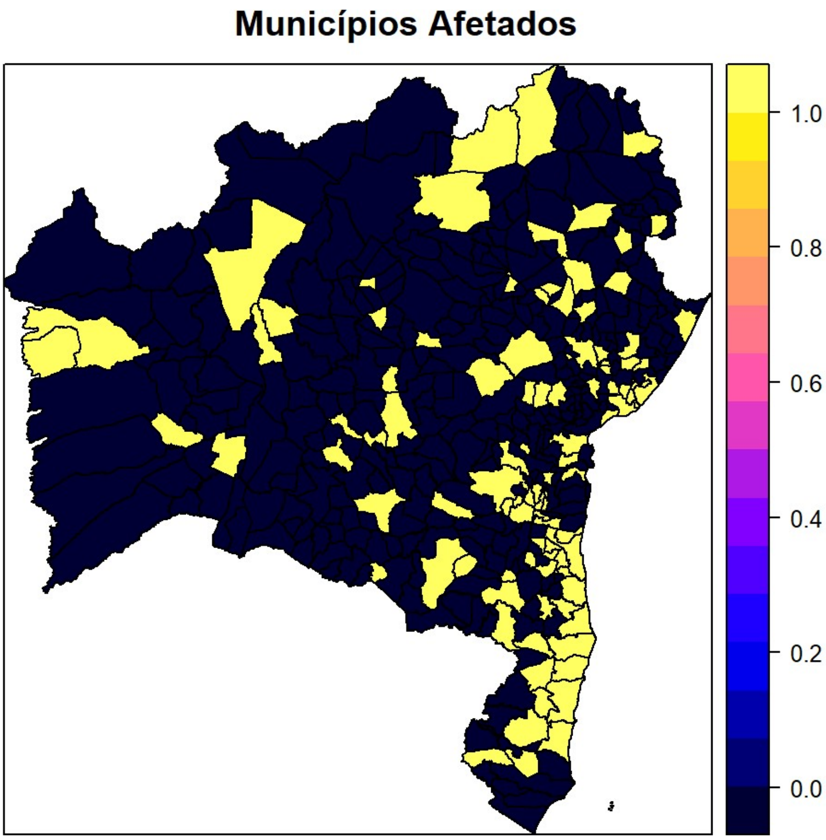

```
sp::spplot(ba.shp, "perc.rural", main="% População Rural")
```




```
sp::spplot(ba.shp, "dist.min.ilh.ssa", main="Distância de Aeroportos")
```

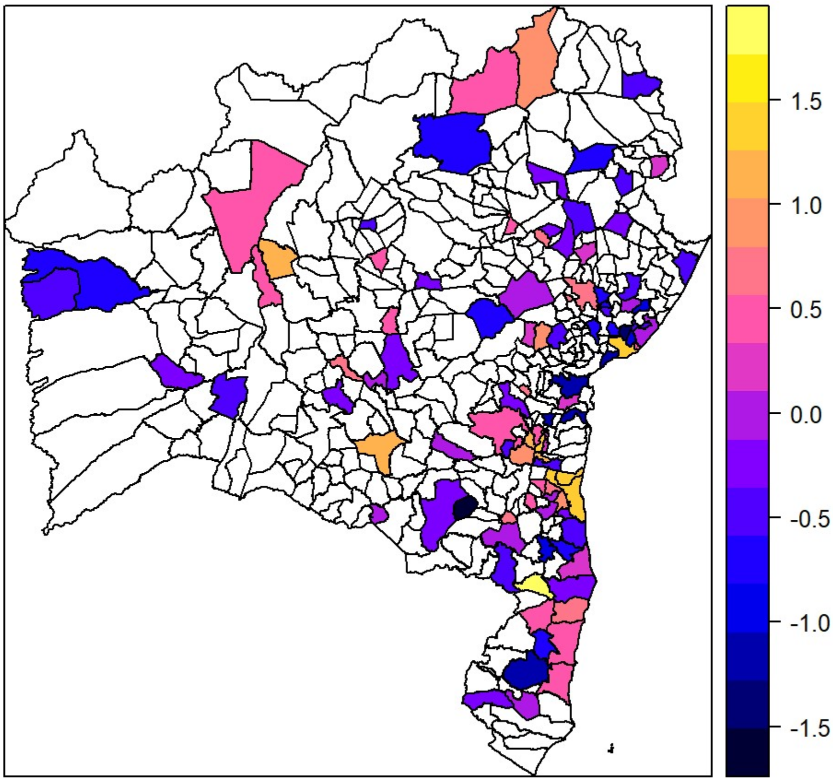


```
sp::spplot(ba.shp, "afetados", main="Municípios Afetados")
```




```
sp::spplot(ba.shp, "res_casos", main="Resíduos Casos")
```

Resíduos Casos




```
sp::spplot(ba.shp, "pred_afetados", main="Probabilidade Afetados")
```

Probabilidade Afetados