

# Caracterização demográfica de cães e gatos domiciliados em municípios brasileiros

Treinamento técnico, 2018

Oswaldo Santos Baquero  
baquero@usp.br  
Laboratório de Epidemiologia e Estatística  
Departamento de Medicina Veterinária Preventiva e Saúde Animal  
Universidade de São Paulo

Exemplo com dados reais de Pinhais, Paraná, descritos em *Baquero, Oswaldo Santos, Solange Marconcini, Adriel Rocha, and Rita de Cassia Maria Garcia. "Companion animal demography and population management in Pinhais, Brazil." Preventive veterinary medicine 158 (2018): 169-177.*

## Pacotes e limpeza da área de trabalho

```
library(gridExtra); library(tidyverse); library(jtools); library(weights);  
library(capm); library(sf); library(ggsn); library(circlize)
```

```
rm(list = ls())  
cat("\014")
```

```
sessionInfo()
```

```
R version 3.4.4 (2018-03-15)  
Platform: x86_64-pc-linux-gnu (64-bit)  
Running under: Ubuntu 17.10
```

```
Matrix products: default  
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1  
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
```

```
locale:  
[1] LC_CTYPE=pt_BR.UTF-8      LC_NUMERIC=C  
[3] LC_TIME=pt_BR.UTF-8      LC_COLLATE=en_US.UTF-8  
[5] LC_MONETARY=pt_BR.UTF-8  LC_MESSAGES=en_US.UTF-8  
[7] LC_PAPER=pt_BR.UTF-8     LC_NAME=C  
[9] LC_ADDRESS=C             LC_TELEPHONE=C  
[11] LC_MEASUREMENT=pt_BR.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:  
[1] grid      stats      graphics  grDevices  utils      datasets  methods  
[8] base
```

```
other attached packages:  
[1] circlize_0.4.5  ggsn_0.4.11    sf_0.7-1      capm_0.13.9  
[5] weights_0.90    mice_3.1.0     gdata_2.18.0  Hmisc_4.1-1  
[9] Formula_1.2-3   survival_2.42-3 lattice_0.20-35 jtools_1.0.0  
[13] forcats_0.3.0   stringr_1.3.1  dplyr_0.7.8   purrr_0.2.5
```

```
[17] readr_1.1.1      tidyr_0.8.2      tibble_1.4.2     ggplot2_3.1.0
[21] tidyverse_1.2.1 gridExtra_2.3
```

loaded via a namespace (and not attached):

```
[1] minqa_1.2.4      colorspace_1.3-2  rjson_0.2.20
[4] class_7.3-14     rprojroot_1.3-2   htmlTable_1.12
[7] GlobalOptions_0.1.0 base64enc_0.1-3    rstudioapi_0.7
[10] lubridate_1.7.4  xml2_1.2.0        splines_3.4.4
[13] mnormt_1.5-5     rootSolve_1.7     knitr_1.20
[16] jsonlite_1.5     nloptr_1.0.4      broom_0.4.5
[19] cluster_2.0.7-1  png_0.1-7         compiler_3.4.4
[22] httr_1.3.1       backports_1.1.2   assertthat_0.2.0
[25] Matrix_1.2-14    lazyeval_0.2.1    survey_3.34
[28] cli_1.0.0        acepack_1.4.1     htmltools_0.3.6
[31] tools_3.4.4      bindrcpp_0.2.2    ggmap_2.7.900
[34] coda_0.19-1      gtable_0.2.0      glue_1.3.0
[37] reshape2_1.4.3   Rcpp_1.0.0        cellranger_1.1.0
[40] nlme_3.1-137     psych_1.8.4       lme4_1.1-17
[43] rvest_0.3.2      gtools_3.8.1      pan_1.6
[46] MASS_7.3-50      scales_1.0.0      hms_0.4.2
[49] parallel_3.4.4   RColorBrewer_1.1-2 yaml_2.1.19
[52] rpart_4.1-13     latticeExtra_0.6-28 stringi_1.2.4
[55] maptools_0.9-2   e1071_1.6-8       checkmate_1.8.5
[58] shape_1.4.4      FME_1.3.5         spData_0.2.9.0
[61] RgoogleMaps_1.4.2 rlang_0.3.0.1     pkgconfig_2.0.2
[64] bitops_1.0-6     evaluate_0.10.1   bindr_0.1.1
[67] htmlwidgets_1.2  tidyselect_0.2.5  deSolve_1.21
[70] plyr_1.8.4       magrittr_1.5      R6_2.3.0
[73] mitml_0.3-6      DBI_1.0.0         pillar_1.2.3
[76] haven_1.1.2      foreign_0.8-71    withr_2.1.2.9000
[79] units_0.6-1      sp_1.3-1          nnet_7.3-12
[82] modelr_0.1.2     crayon_1.3.4      jomo_2.6-2
[85] rmarkdown_1.10   jpeg_0.1-8        readxl_1.1.0
[88] minpack.lm_1.2-1 data.table_1.11.4 digest_0.6.18
[91] classInt_0.2-3   munsell_0.5.0
```

## Amostragem

```
## Dados do IBGE
basico <- read_csv2("Basico_PR.csv", locale = locale(encoding = "latin1"))
psu_ssu <- basico %>%
  filter(Nome_do_municipio == 'PINHAIS') %>%
  select(Cod_setor)
doms <- read_csv2("Domicilio01_PR.csv", locale = locale(encoding = "latin1"))
psu_ssu <- merge(psu_ssu, doms[, c(1, 3)], by = 'Cod_setor')
write_csv(psu_ssu, "psu_ssu.csv")

pinhais <- read_sf("./mapa_parana/41MUE250GC_SIR.shp")
pinhais <- pinhais %>%
  filter(NM_MUNICIP == "PINHAIS") %>%
  transmute(Cod_setor = as.numeric(CD_GEOCODI)) %>%
  left_join(psu_ssu, by = "Cod_setor")
```

```
## Unidades amostrais
psu <- SamplePPS(psu.ssu = psu_ssu, psu = 45)
ssu <- SampleSystematic(psu.ssu = psu, su = 30, write = TRUE)

## Mapas dos setores selecionados
MapkmlPSU(shape = pinhais, psu = psu, id = 1)
```

## Dados das entrevistas

```
banco1 <- read_csv("banco1.csv")
```

```
Parsed with column specification:
cols(
  ID = col_integer(),
  codigo_do_setor_censitario = col_double(),
  entrevistador = col_character(),
  data = col_date(format = ""),
  endereco = col_character(),
  entrevista = col_character(),
  entrevistado = col_character(),
  quantos_caes = col_integer(),
  quantos_gatos = col_integer(),
  quantas_pessoas = col_integer(),
  razao_para_nao_castrar = col_character(),
  razao_para_nao_castrar_outras = col_character()
)
```

```
psu_ssu <- read_csv("psu_ssu.csv")
```

```
Parsed with column specification:
cols(
  Cod_setor = col_double(),
  V001 = col_integer()
)
```

```
banco2 <- read_csv("banco2.csv")
```

```
Parsed with column specification:
cols(
  .default = col_character(),
  ID = col_integer(),
  codigo_do_setor_censitario = col_double(),
  idade = col_integer(),
  filhotes_u12 = col_integer(),
  idade3 = col_integer()
)
```

See `spec(...)` for full column specifications.

```
dogs <- filter(banco2, especie == "cao" | especie3 == "cao")
cats <- filter(banco2, especie == "gato" | especie3 == "gato")
```

## Status da entrevista

```
FreqTab(banco1$entrevista)
```

	Category	Count	Proportion
1	atendida	967	0.786
2	recusa	196	0.159
3	casa_fechada	67	0.054

## Estimativas

### População humana em 2017

```
pinhais2017 <- 129445
```

## Presença de Cães e gatos nos domicílios

```
names(banco1)
```

```
[1] "ID" "codigo_do_setor_censitario"
[3] "entrevistador" "data"
[5] "endereco" "entrevista"
[7] "entrevistado" "quantos_caes"
[9] "quantos_gatos" "quantas_pessoas"
[11] "razao_para_nao_castrar" "razao_para_nao_castrar_outras"
```

```
banco1$tem_caes <- ifelse(banco1$quantos_caes > 0, 1, 0)
banco1$tem_gatos <- ifelse(banco1$quantos_gatos > 0, 1, 0)
```

## Seleção das variáveis de interesse e remoção de observações com informações faltantes

```
banco1_comp <- banco1 %>%
  select(ID, codigo_do_setor_censitario, quantos_caes,
         quantos_gatos, quantas_pessoas, tem_caes, tem_gatos) %>%
  filter(complete.cases(.))
```

## Desenhos amostrais

```
design <- DesignSurvey(banco1_comp, psu.ssu = psu_ssu,
                      psu.col = "codigo_do_setor_censitario",
                      ssu.col = "ID")
design_cal <- DesignSurvey(banco1_comp, psu.ssu = psu_ssu,
                          psu.col = "codigo_do_setor_censitario",
                          ssu.col = "ID",
                          cal.col = "quantas_pessoas",
```

```

cal.N = pinhais2017)
summary(design)

2 - level Cluster Sampling design
With (40, 967) clusters.
svydesign(ids = ~psu.id + ssu.id, fpc = ~pop.size + psu.size,
  weights = ~weights, data = sample, ...)
Probabilities:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01292 0.02353 0.02654 0.02903 0.03457 0.05139
Population size (PSUs): 136
Data variables:
[1] "quantos_caes"    "quantos_gatos"    "quantas_pessoas" "tem_caes"
[5] "tem_gatos"
summary(design_cal)

```

```

2 - level Cluster Sampling design
With (40, 967) clusters.
calibrate(dsn, formula = ~sample[, cal.col] - 1, population = cal.N)
Probabilities:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01143 0.02164 0.02504 0.02709 0.03244 0.04924
Population size (PSUs): 136
Data variables:
[1] "quantos_caes"    "quantos_gatos"    "quantas_pessoas" "tem_caes"
[5] "tem_gatos"

```

## Resultados

```
(est <- SummarySurvey(design, c(rep("total", 3), "prop", "prop")))
```

	Estimate	SE	2.5 %	97.5 %	Deff
Total_quantos_caes	46874.032	2526.973	41921.256	51826.807	2.322
Total_quantos_gatos	7191.936	964.076	5302.382	9081.490	1.757
Total_quantas_pessoas	119120.826	4665.893	109975.843	128265.809	7.389
Prop_tem_caes	0.665	0.018	0.629	0.701	1.506
Prop_tem_gatos	0.126	0.013	0.100	0.152	1.615
	Error (%)				
Total_quantos_caes	10.566				
Total_quantos_gatos	26.274				
Total_quantas_pessoas	7.677				
Prop_tem_caes	5.423				
Prop_tem_gatos	20.811				

```
(est_cal <- SummarySurvey(design_cal, c(rep("total", 3), "prop", "prop")))
```

	Estimate	SE	2.5 %	97.5 %	Deff
Total_quantos_caes	50443.792	2148.924	46231.979	54655.606	1.456
Total_quantos_gatos	7721.507	1008.150	5745.568	9697.445	1.673
Total_quantas_pessoas	129445.000	0.000	129445.000	129445.000	0.000
Prop_tem_caes	0.667	0.018	0.631	0.702	1.447
Prop_tem_gatos	0.127	0.014	0.100	0.153	1.635
	Error (%)				

Total_quantos_caes	8.350
Total_quantos_gatos	25.591
Total_quantas_pessoas	0.000
Prop_tem_caes	5.299
Prop_tem_gatos	20.908

## Outros dados amostrais

### Correlação pessoas x cães

```
cor.test(banco1_comp$quantas_pessoas, banco1_comp$quantos_caes)
```

Pearson's product-moment correlation

data: banco1\_comp\$quantas\_pessoas and banco1\_comp\$quantos\_caes

t = 3.6356, df = 965, p-value = 0.000292

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.05359005 0.17797805

sample estimates:

cor

0.1162398

*# Levando em conta o desenho amostral*

```
h_d_cor <- svycor(~quantas_pessoas + quantos_caes, design_cal, sig.stats = TRUE)
```

```
h_d_cor$cors[1, 2]
```

```
[1] 0.1215061
```

```
h_d_cor$cors[1, 2] - h_d_cor$std.err[1, 2] * 1.96
```

```
[1] 0.05653324
```

```
h_d_cor$cors[1, 2] + h_d_cor$std.err[1, 2] * 1.96
```

```
[1] 0.1864789
```

### Correlação pessoas x gatos

```
cor.test(banco1_comp$quantas_pessoas, banco1_comp$quantos_gatos)
```

Pearson's product-moment correlation

data: banco1\_comp\$quantas\_pessoas and banco1\_comp\$quantos\_gatos

t = 0.64088, df = 965, p-value = 0.5218

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.04247134 0.08356018

sample estimates:

cor

0.02062636

```
# Levando em conta o desenho amostral
h_c_cor <- svycor(~quantas_pessoas + quantos_gatos, design_cal, sig.stats = TRUE)
h_c_cor$cors[1, 2]
```

```
[1] 0.01790971
```

```
h_c_cor$cors[1, 2] - h_c_cor$std.err[1, 2] * 1.96
```

```
[1] -0.04524955
```

```
h_c_cor$cors[1, 2] + h_c_cor$std.err[1, 2] * 1.96
```

```
[1] 0.08106896
```

## Cães, gatos e pessoas por domicilio

```
summary(banco1$quantos_caes)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.000	0.000	1.000	1.301	2.000	12.000	263

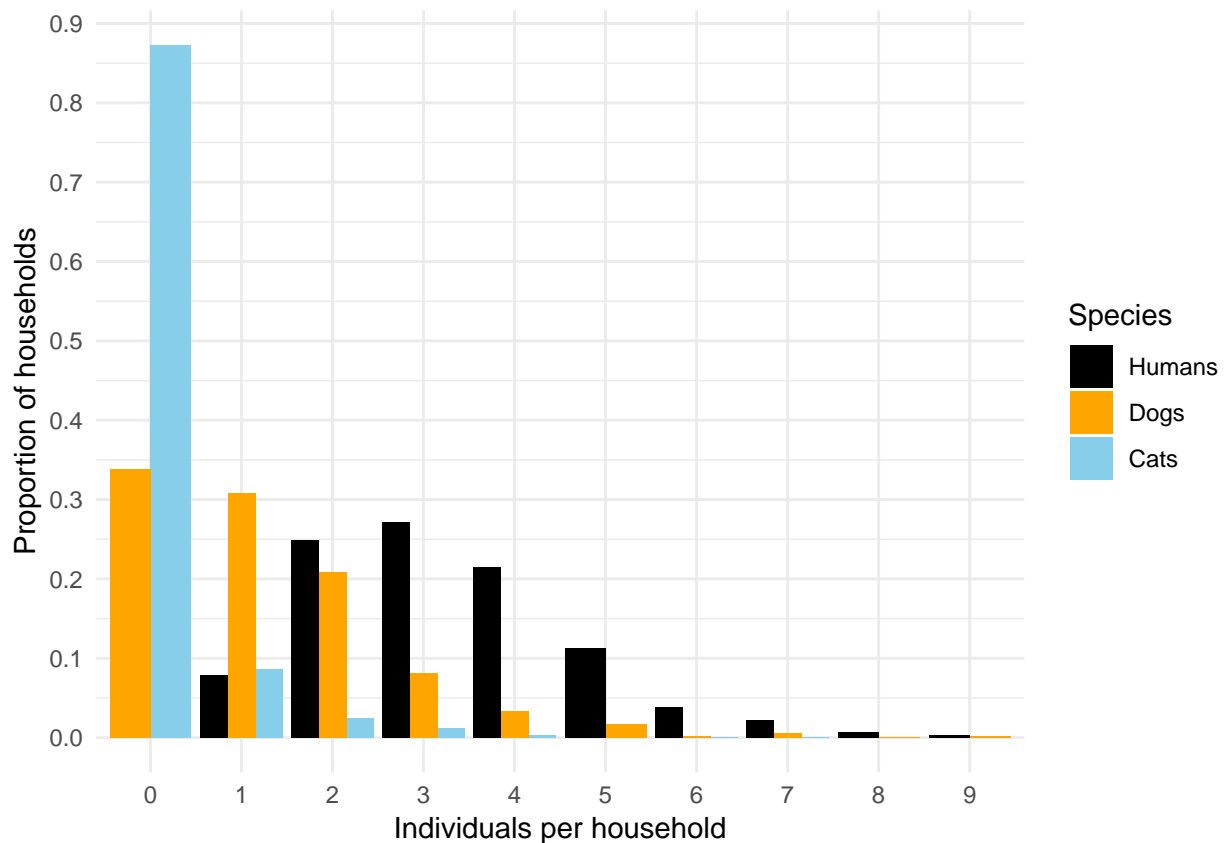
```
summary(banco1$quantos_gatos)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.0000	0.0000	0.0000	0.1965	0.0000	7.0000	263

```
summary(banco1$quantas_pessoas)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
1.000	2.000	3.000	3.306	4.000	15.000	263

```
#tiff("./fig2.tiff", width = 2e3, height = 2e3, res = 300)
PlotHHxSpecies(banco1_comp, c("quantos_caes", "quantos_gatos", "quantas_pessoas")) +
  scale_fill_manual(values = c("black", "orange", "skyblue"),
                    labels = c("Humans", "Dogs", "Cats")) +
  scale_y_continuous(breaks = seq(0, .9, by = .1)) +
  xlim(as.character(0:9))
```



```
#dev.off()
```

## Razão de sexos

### Cães

```
FreqTab(c(dogs$sexo, dogs$sexo3))
```

Category	Count	Proportion
1 macho	644	0.515
2 fema	606	0.485

```
prop_sex <- table(c(dogs$sexo, dogs$sexo3))
prop.test(prop_sex["femea"], sum(prop_sex), p = .5)
```

1-sample proportions test with continuity correction

```
data: prop_sex["femea"] out of sum(prop_sex), null probability 0.5
X-squared = 1.0952, df = 1, p-value = 0.2953
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4567856 0.5129088
sample estimates:
p
0.4848
```



## Gatos

```
FreqTab(c(cats$sexo, cats$sexo3))
```

	Category	Count	Proportion
1	femea	110	0.564
2	macho	85	0.436

```
prop_sex_c <- table(c(cats$sexo, cats$sexo3))
prop.test(prop_sex_c["femea"], sum(prop_sex_c), p = .5)
```

1-sample proportions test with continuity correction

```
data: prop_sex_c["femea"] out of sum(prop_sex_c), null probability 0.5
X-squared = 2.9538, df = 1, p-value = 0.08567
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4913709 0.6342670
sample estimates:
      p
0.5641026
```

## Esterilização e sexo

### Cães

```
(prop_sex_ster <- xtabs(~ c(dogs$castrado, dogs$castrado3) +
                          c(dogs$sexo, dogs$sexo3)))
```

	c(dogs\$sexo, dogs\$sexo3)	
c(dogs\$castrado, dogs\$castrado3)	femea	macho
nao	406	515
sim	191	120

```
prop.test(prop_sex_ster["sim", ], apply(prop_sex_ster, 2, sum))
```

2-sample test for equality of proportions with continuity correction

```
data: prop_sex_ster["sim", ] out of apply(prop_sex_ster, 2, sum)
X-squared = 27.274, df = 1, p-value = 1.766e-07
alternative hypothesis: two.sided
95 percent confidence interval:
 0.0810907 0.1808225
sample estimates:
 prop 1    prop 2
0.3199330 0.1889764
```

## Gatos

```
(prop_sex_ster_c <- xtabs(~ c(cats$castrado, cats$castrado3) +  
                           c(cats$sexo, cats$sexo3)))  
  
                           c(cats$sexo, cats$sexo3)  
c(cats$castrado, cats$castrado3) femea macho  
      nao      68      53  
      sim      38      31  
  
prop.test(prop_sex_ster_c["sim", ], apply(prop_sex_ster_c, 2, sum))
```

2-sample test for equality of proportions with continuity  
correction

```
data: prop_sex_ster_c["sim", ] out of apply(prop_sex_ster_c, 2, sum)  
X-squared = 1.186e-30, df = 1, p-value = 1  
alternative hypothesis: two.sided  
95 percent confidence interval:  
 -0.1588930  0.1377789  
sample estimates:  
   prop 1    prop 2  
0.3584906 0.3690476
```

## Status semi-domiciliado e sexo

### Cães

```
(prop_sex_fr <- xtabs(~ dogs$sai_sozinho_a_rua + dogs$sexo))  
  
               dogs$sexo  
dogs$sai_sozinho_a_rua femea macho  
      nao      455      428  
      sim       73      114  
  
prop.test(prop_sex_fr["sim", ], apply(prop_sex_fr, 2, sum))
```

2-sample test for equality of proportions with continuity  
correction

```
data: prop_sex_fr["sim", ] out of apply(prop_sex_fr, 2, sum)  
X-squared = 9.1401, df = 1, p-value = 0.002501  
alternative hypothesis: two.sided  
95 percent confidence interval:  
 -0.11915474 -0.02499431  
sample estimates:  
   prop 1    prop 2  
0.1382576 0.2103321
```

## Gatos

```
(prop_sex_fr_c <- xtabs(~ cats$sai_sozinho_a_rua + cats$sexo))
```

```
      cats$sexo  
cats$sai_sozinho_a_rua femea macho  
      nao      65      30  
      sim      31      42
```

```
prop.test(prop_sex_fr_c["sim", ], apply(prop_sex_fr_c, 2, sum))
```

2-sample test for equality of proportions with continuity  
correction

```
data: prop_sex_fr_c["sim", ] out of apply(prop_sex_fr_c, 2, sum)  
X-squared = 10.32, df = 1, p-value = 0.001316  
alternative hypothesis: two.sided  
95 percent confidence interval:  
 -0.4199359 -0.1008974  
sample estimates:  
   prop 1   prop 2  
0.3229167 0.5833333
```

## Status semi-domiciliado e esterilizacao

### Cães

```
(prop_sex_ster <- xtabs(~ dogs$sai_sozinho_a_rua + dogs$castrado))
```

```
      dogs$castrado  
dogs$sai_sozinho_a_rua nao sim  
      nao 625 245  
      sim 151 36
```

```
prop.test(prop_sex_ster["sim", ], apply(prop_sex_ster, 2, sum))
```

2-sample test for equality of proportions with continuity  
correction

```
data: prop_sex_ster["sim", ] out of apply(prop_sex_ster, 2, sum)  
X-squared = 5.8119, df = 1, p-value = 0.01592  
alternative hypothesis: two.sided  
95 percent confidence interval:  
 0.01606198 0.11688552  
sample estimates:  
   prop 1   prop 2  
0.1945876 0.1281139
```

## Gatos

```
(prop_sex_ster_c <- xtabs(~ cats$sai_sozinho_a_rua + cats$castrado))
```

```
      cats$castrado
cats$sai_sozinho_a_rua nao sim
      nao  56  36
      sim  43  29
```

```
prop.test(prop_sex_ster_c["sim", ], apply(prop_sex_ster_c, 2, sum))
```

2-sample test for equality of proportions with continuity correction

```
data: prop_sex_ster_c["sim", ] out of apply(prop_sex_ster_c, 2, sum)
X-squared = 9.0023e-31, df = 1, p-value = 1
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.1789813  0.1553605
sample estimates:
 prop 1    prop 2 
0.4343434 0.4461538
```

## Idade

```
summary(c(dogs$idade, dogs$idade3))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.000	1.000	4.000	5.086	8.000	25.000	1277

```
summary(c(cats$idade, cats$idade3))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	0.00	2.00	2.77	4.00	15.00	203

## Pirâmides populacionais

### Cães

```
pp_age <- c(dogs$idade, dogs$idade3)
levels(as.factor(dogs$sexo))
```

```
[1] "femea" "macho"
```

```
pp_sex <- factor(c(dogs$sexo, dogs$sexo3),
               levels = c("femea", "macho"),
               labels = c("Female", "Male"))
levels(as.factor(dogs$castrado))
```

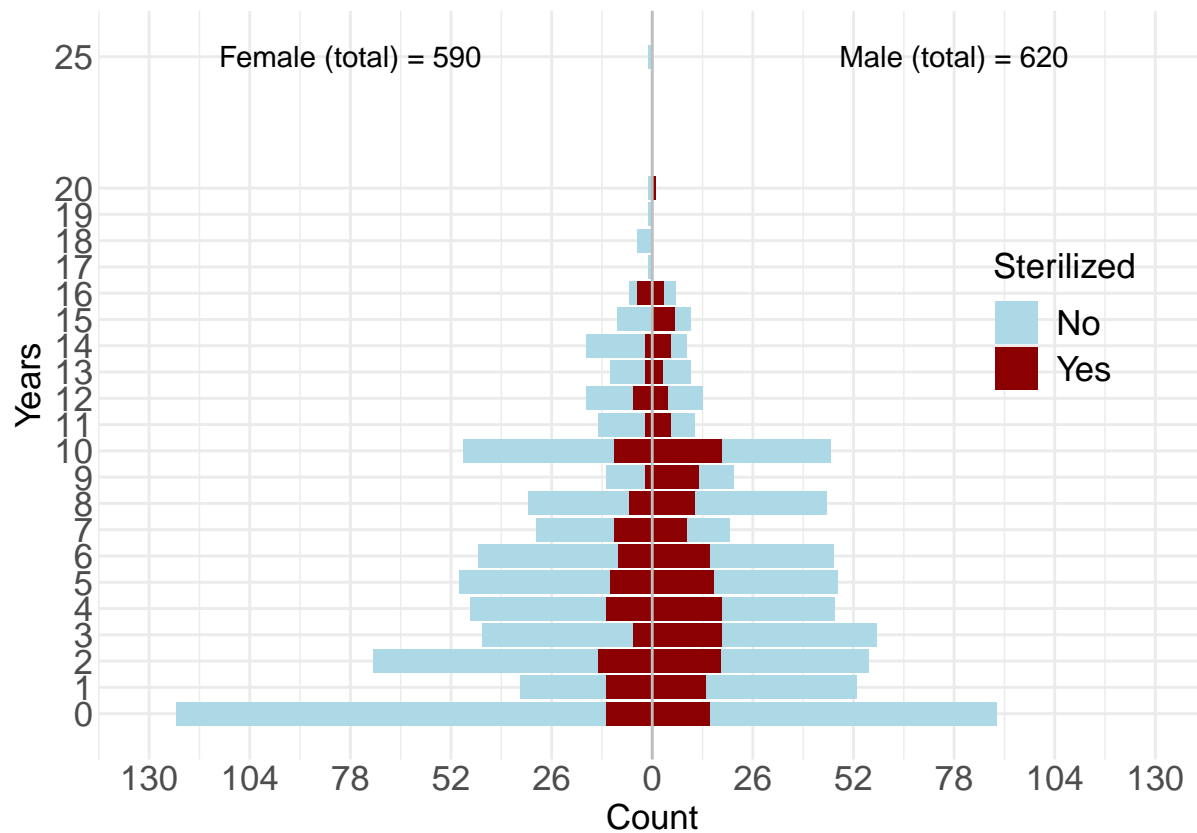
```
[1] "nao" "sim"
```

```
pp_ster <- factor(c(dogs$castrado, dogs$castrado3),
                 levels = c("nao", "sim"),
```

```

labels = c("No", "Yes"))
pop_pir <- data.frame(pp_age, pp_sex, pp_ster)
PlotPopPyramid(pop_pir, "pp_age", "pp_sex", "pp_ster") +
  theme(legend.position = c(.95, .7))

```

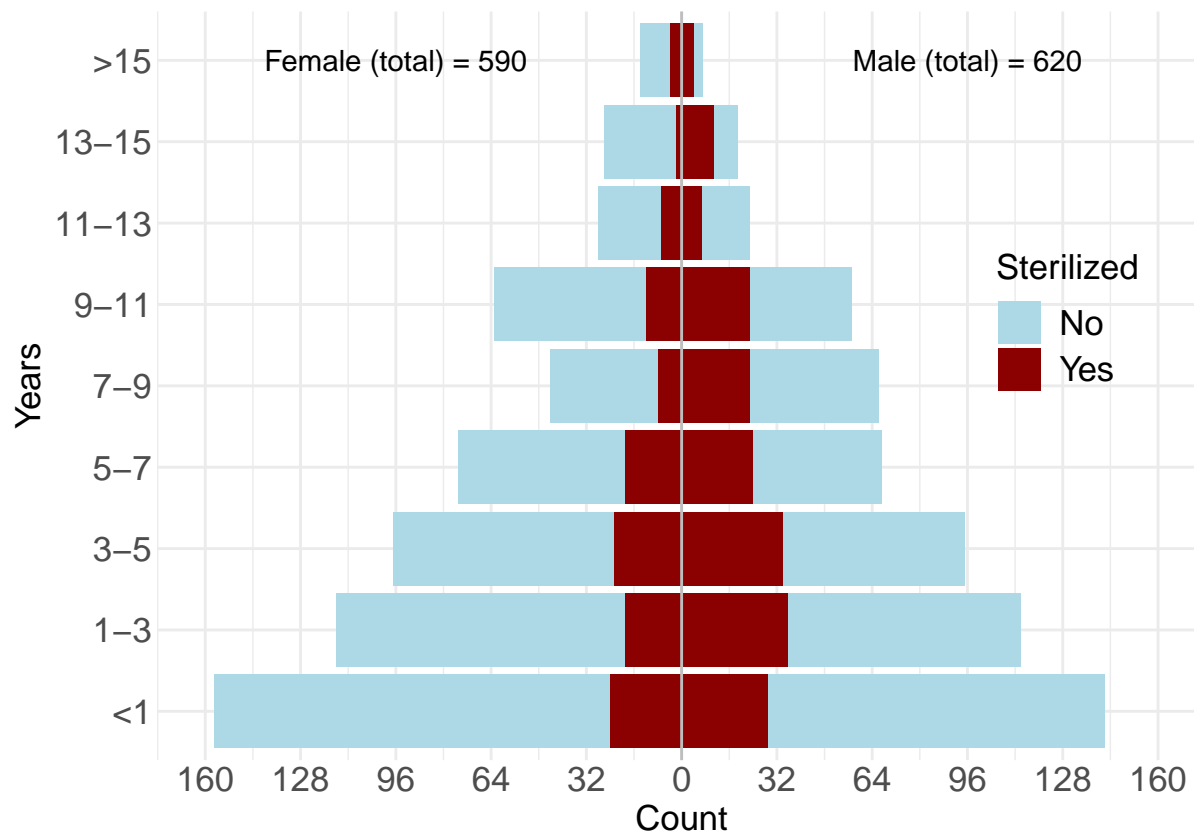


```

pp_age2 <- cut(c(dogs$idade, dogs$idade3),
  breaks = c(0, 1, 3, 5, 7, 9, 11, 13, 15,
    max(pp_age, na.rm = TRUE)),
  labels = c("<1", "1-3", "3-5", "5-7", "7-9",
    "9-11", "11-13", "13-15", ">15"),
  include.lowest = TRUE)
pop_pir2 <- data.frame(pp_age2, pp_sex, pp_ster)

#tiff("./fig3a.tiff", width = 2e3, height = 2e3, res = 300)
PlotPopPyramid(pop_pir2, "pp_age2", "pp_sex", "pp_ster",
  inner.color = "LightBlue",
  outer.color = "DarkRed") +
  theme(legend.position = c(.95, .7))

```



```
#dev.off()
```

## Gatos

```
pp_age_c <- c(cats$idade, cats$idade3)
levels(as.factor(cats$sexo))

[1] "femea" "macho"

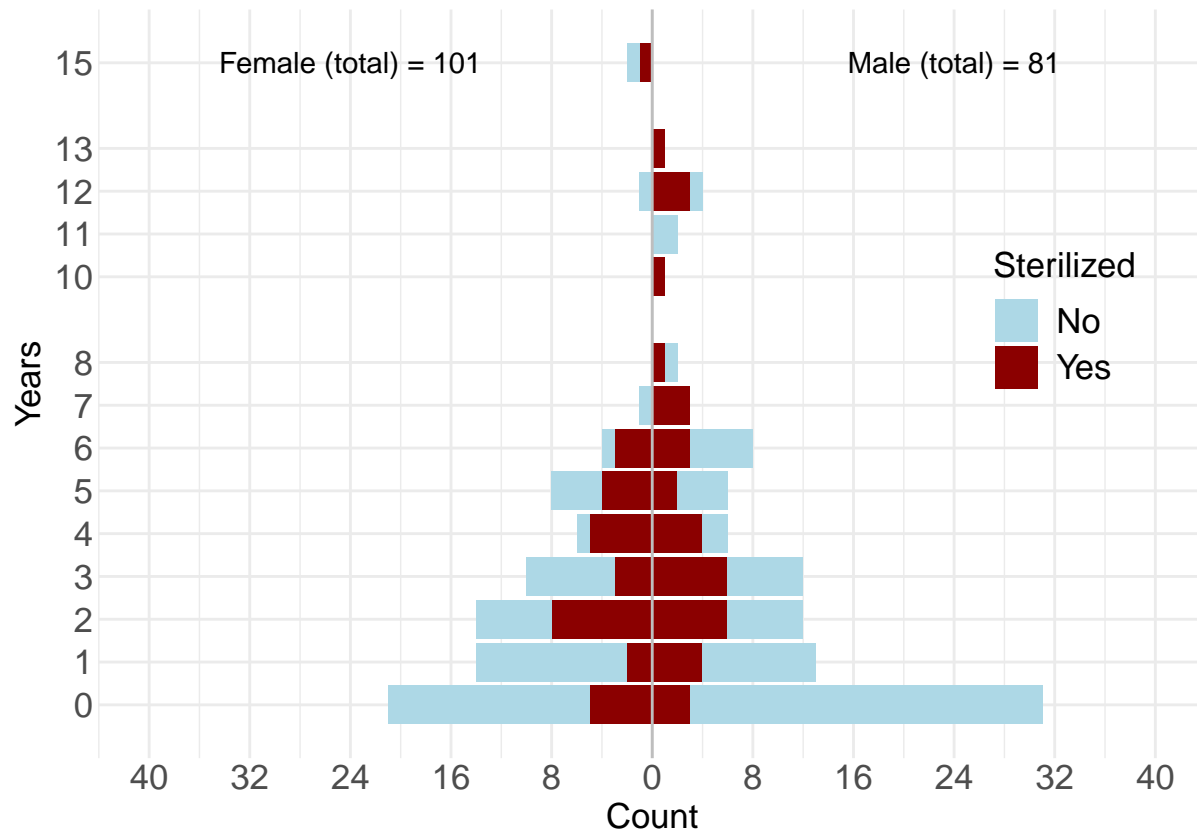
pp_sex_c <- factor(c(cats$sexo, cats$sexo3),
  levels = c("femea", "macho"),
  labels = c("Female", "Male"))
levels(as.factor(cats$castrado))

[1] "nao" "sim"

pp_ster_c <- factor(c(cats$castrado, cats$castrado3),
  levels = c("nao", "sim"),
  labels = c("No", "Yes"))

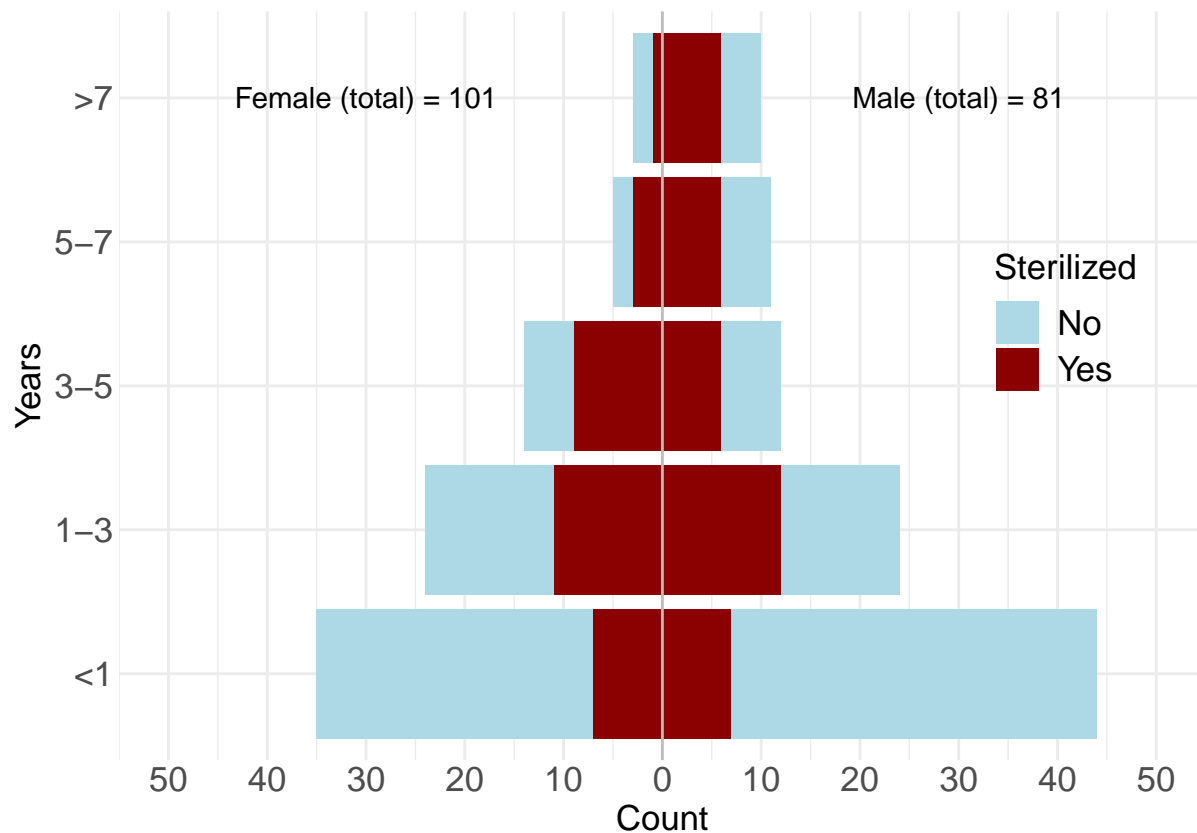
pop_pir_c <- data.frame(pp_age_c, pp_sex_c, pp_ster_c)

PlotPopPyramid(pop_pir_c, "pp_age_c", "pp_sex_c", "pp_ster_c") +
  theme(legend.position = c(.95, .7))
```



```
pp_age2_c <- cut(c(cats$idade, cats$idade3),
  breaks = c(0, 1, 3, 5, 7, max(pp_age_c, na.rm = TRUE)),
  labels = c("<1", "1-3", "3-5", "5-7", ">7"),
  include.lowest = TRUE)
pop_pir2_c <- data.frame(pp_age2_c, pp_sex_c, pp_ster_c)

#tiff("./fig3b.tiff", width = 2e3, height = 2e3, res = 300)
PlotPopPyramid(pop_pir2_c, "pp_age2_c", "pp_sex_c", "pp_ster_c",
  inner.color = "LightBlue",
  outer.color = "DarkRed") +
  theme(legend.position = c(.95, .7))
```



```
#dev.off()
```

## Tamanho das ninhadas

```
summary(dogs$filhotes_u12[dogs$filhotes_u12 > 0])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
1.000	4.000	5.000	5.281	6.000	16.000	608

```
summary(cats$filhotes_u12[cats$filhotes_u12 > 0])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
1.000	1.250	2.500	2.667	3.750	5.000	86

## Proporção de fêmeas que tiveram cria

```
sum(dogs$filhotes_u12 > 0, na.rm = T) / sum(dogs$sexo == "femea", na.rm = T)
```

```
[1] 0.06060606
```

```
sum(cats$filhotes_u12 > 0, na.rm = T) / sum(cats$sexo == "femea", na.rm = T)
```

```
[1] 0.0625
```



## Proporção de filhotes em relação ao total da população

```
sum(dogs$filhotes_u12, na.rm = TRUE) / nrow(dogs)
```

```
[1] 0.134984
```

```
sum(cats$filhotes_u12, na.rm = TRUE) / nrow(cats)
```

```
[1] 0.08205128
```

## Aquisição

```
FreqTab(dogs$aquisicao)
```

	Category	Count	Proportion
1	adotou	536	0.509
2	ganhou	401	0.380
3	comprou	117	0.111

```
FreqTab(cats$aquisicao)
```

	Category	Count	Proportion
1	adotou	119	0.721
2	ganhou	42	0.255
3	comprou	4	0.024

## Cidade de aquisição

### Cães

#### Frequencias por estado

```
FreqTab(dogs$estado_de_aquisicao)
```

	Category	Count	Proportion
1	parana	1239	0.990
2	santa catarina	7	0.006
3	sao paulo	2	0.002
4	distrito federal	1	0.001
5	mato grosso	1	0.001
6	portugal	1	0.001
7	rio grande do sul	1	0.001

## Número de imigrantes por cidade de origem

```
(cities <- cities2 <- FreqTab(dogs$municipio_de_aquisicao))
```

	Category	Count	Proportion
1	Pinhais	874	0.817
2	Curitiba	100	0.093
3	Piraquara	21	0.020
4	Colombo	15	0.014

5	Sao Jose dos Pinhais	12	0.011
6	Campo Largo	5	0.005
7	Quatro Barras	4	0.004
8	Araucaria	3	0.003
9	Joinville	3	0.003
10	Cascavel	2	0.002
11	Nao Sabe	2	0.002
12	Paranagua	2	0.002
13	Tijucas Do Sul	2	0.002
14	Agua Clara	1	0.001
15	Almirante Tamandare	1	0.001
16	Arapongas	1	0.001
17	Argentina	1	0.001
18	Bocaiuva	1	0.001
19	Cambara	1	0.001
20	Campina Grande Do Sul	1	0.001
21	Canoas	1	0.001
22	Capanema	1	0.001
23	Cic	1	0.001
24	Fazenda Rio Grande	1	0.001
25	Itajai	1	0.001
26	Itapema	1	0.001
27	Litoral	1	0.001
28	Londrina	1	0.001
29	Mato Grosso	1	0.001
30	Norte Do Parana	1	0.001
31	Ortigueira	1	0.001
32	Ponta Grossa	1	0.001
33	Portugal	1	0.001
34	Presidente Prudente	1	0.001
35	Santa Catarina	1	0.001
36	Sao Paulo	1	0.001
37	Telemaco Borba	1	0.001
38	Treze Tilias	1	0.001

Número de imigrantes por cidade de origem (versao para graficar)

```
cities <- cities[, c(1, 3)]
names(cities)[1] <- "NM_MUNICIP"
```

Mapa com as cidades de Paraná

```
pr <- read_sf("./mapa_parana/41MUE250GC_SIR.shp", options = "ENCODING=windows-1252")
names(pr)
```

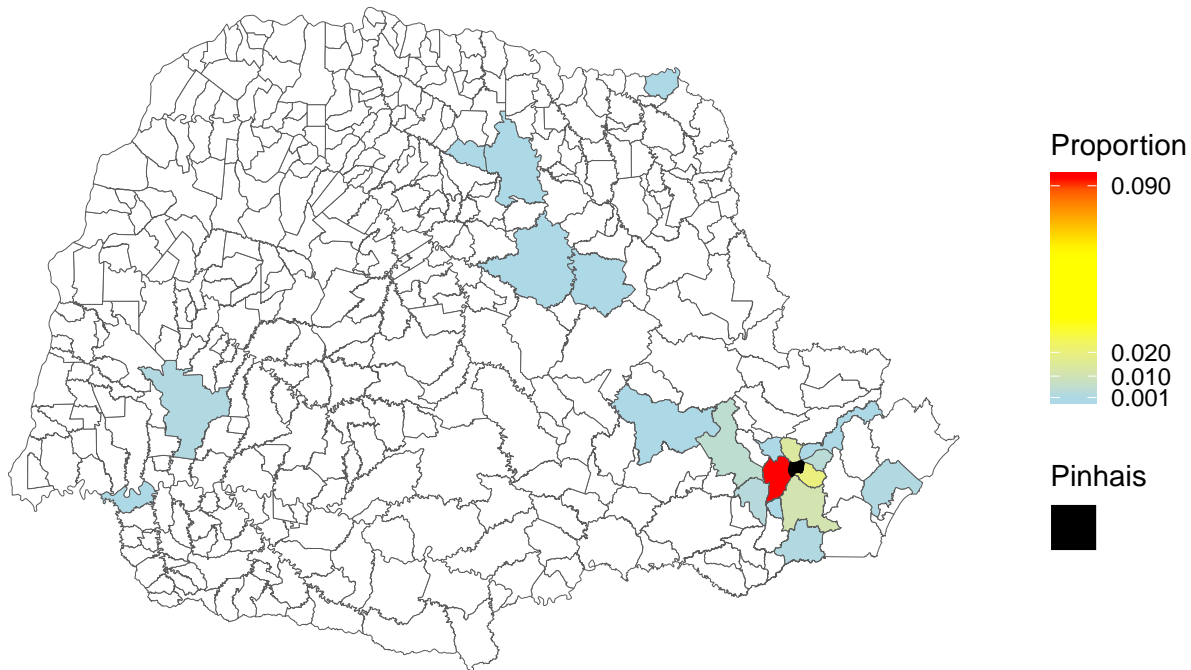
```
[1] "ID" "CD_GEOCODM" "NM_MUNICIP" "geometry"
pr$NM_MUNICIP <- str_to_title(iconv(pr$NM_MUNICIP, to='ASCII//TRANSLIT'))
pr$NM_MUNICIP <- gsub("Dos", "dos", pr$NM_MUNICIP)
pr <- left_join(pr, cities)
```

Joining, by = "NM\_MUNICIP"

Warning: Column `NM\_MUNICIP` joining character vector and factor, coercing into character vector

```
pr[pr$NM_MUNICIP == "Pinhais", "Proportion"] <- NA

(map1 <- ggplot(data = pr, aes(fill = Proportion)) +
  geom_sf(size = .08) +
  geom_sf(data = pr[pr$NM_MUNICIP == "Pinhais", ],
    aes(color = ""), fill = "black") +
  scale_fill_continuous(low = c("lightblue", "yellow"),
    high = c("yellow", "red"),
    na.value = "white", breaks = c(.001, .01, .02, .09)) +
  scale_color_manual(name = "Pinhais", values = "black") +
  blank())
```



## Gatos

### Frequencias por estado de aquisição

```
FreqTab(cats$estado_de_aquisicao)
```

	Category	Count	Proportion
1	parana	193	0.990
2	santa catarina	1	0.005
3	sao paulo	1	0.005

### Número de imigrantes por cidade de origem

```
(cities_c <- cities2_c <- FreqTab(cats$municipio_de_aquisicao))
```

	Category	Count	Proportion
--	----------	-------	------------

1	Pinhais	144	0.857
2	Curitiba	15	0.089
3	Piraquara	2	0.012
4	Almirante Tamandare	1	0.006
5	Colombo	1	0.006
6	Joinville	1	0.006
7	Matinhos	1	0.006
8	Norte Do Pr	1	0.006
9	Sao Jose dos Pinhais	1	0.006
10	Sao Paulo	1	0.006

Número de imigrantes por cidade de origem (versão para graficar)

```
cities_c <- cities_c[, c(1, 3)]
names(cities_c)[1] <- "NM_MUNICIP"
```

## Mapa do Brasil

```
brazil <- read_sf("./mapa_brasil/BRUFE250GC_SIR.shp") %>%
  mutate(NM_ESTADO = ifelse(NM_ESTADO == "PARANÁ", "Paraná", "Brazil"))
#tiff("./fig4a.tiff", width = 4e3, height = 1e3, res = 300)
ggplot() +
  geom_sf(data = brazil, aes(fill = NM_ESTADO), color = "gray") +
  scale_fill_manual(name = "",
                    values = c("Brazil" = "gray", "Paraná" = "orange")) +
  blank() +
  scalebar(brazil, dist = 500, dd2km = TRUE, model = "WGS84", st.size = 3,
            anchor = c(x = -35, y = -32))
```



```
#dev.off()
```

## Mapa das cidades de Paraná

```
pr_c <- read_sf("./mapa_parana/41MUE250GC_SIR.shp", options = "ENCODING=windows-1252")
pr_c$NM_MUNICIP <- str_to_title(iconv(pr_c$NM_MUNICIP, to='ASCII//TRANSLIT'))
pr_c$NM_MUNICIP <- gsub("Dos", "dos", pr_c$NM_MUNICIP)
pr_c <- left_join(pr_c, cities_c)
```

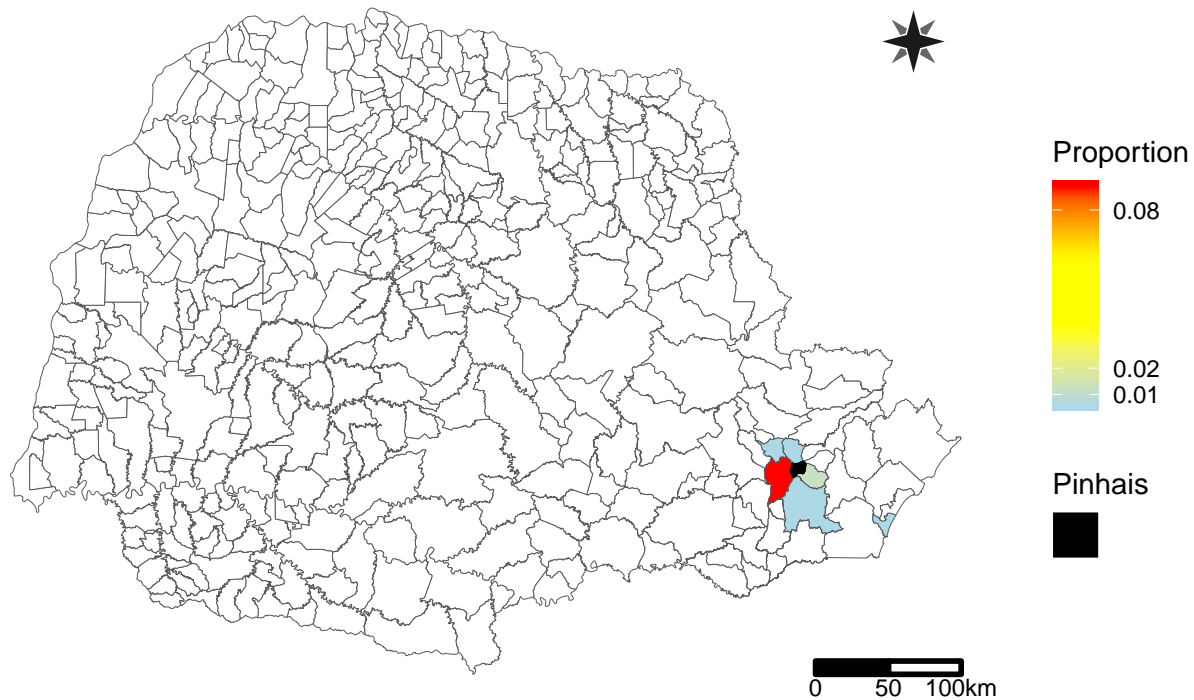
Joining, by = "NM\_MUNICIP"

Warning: Column `NM\_MUNICIP` joining character vector and factor, coercing into character vector

```
pr_c[pr_c$NM_MUNICIP == "Pinhais", "Proportion"] <- NA
```

```
(map2 <- ggplot(data = pr_c, aes(fill = Proportion)) +
  geom_sf(size = .08) +
  geom_sf(data = pr_c[pr_c$NM_MUNICIP == "Pinhais", ],
    aes(color = ""), fill = "black") +
  guides(color = guide_legend(order = 2),
    fill = guide_colorbar(order = 1)) +
  scale_fill_continuous(low = c("lightblue", "yellow"),
    high = c("yellow", "red"),
    na.value = "white", breaks = c(.001, .01, .02, .08)) +
  scale_color_manual(name = "Pinhais", values = "black") +
  blank() +
```

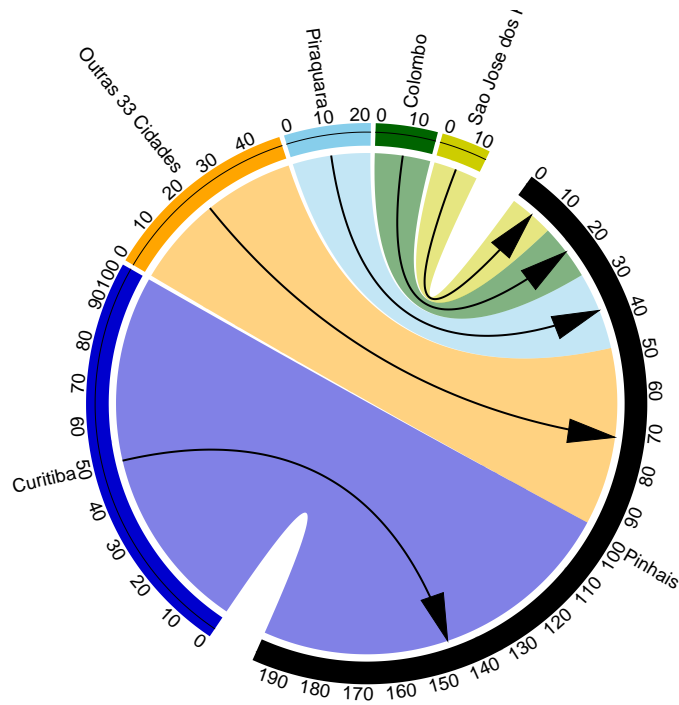
```
scalebar(pr, dist = 50, dd2km = TRUE, model = "WGS84", st.size = 3) +
north(pr, symbol = 15))
```



## Diagramas de cordas

### Cães

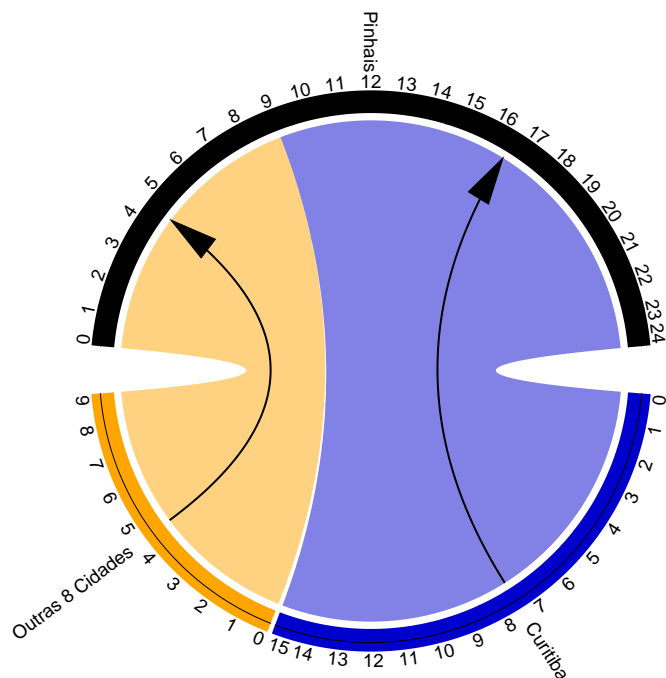
```
cls <- c("blue3", "orange", "skyblue", "darkgreen", "yellow3", "black")
#tiff("./fig5.tiff", width = 1.5e3, height = 3e3, res = 250)
#par(mfrow = c(2, 1), mar = c(0, 0, 0, 0))
PlotImmigrationFlow(dogs,
  source = "municipio_de_aquisicao",
  destination = "Pinhais",
  agg.sources.prefix = "Outras ",
  agg.sources.suffix = " Cidades",
  cls = cls,
  start.degree = 240,
  sources.label.size = .6,
  ticks.label.size = .55)
```



```
#dev.off()
```

## Gatos

```
cls_c <- c("blue3", "orange", "black")
#tiff("./fig6.tiff", width = 1.5e3, height = 3e3, res = 250)
#par(mfrow = c(2, 1), mar = c(0, 0, 0, 0))
PlotImmigrationFlow(cats,
  n.sources = 2,
  source = "municipio_de_aquisicao",
  destination = "Pinhais",
  agg.sources.prefix = "Outras ",
  agg.sources.suffix = " Cidades",
  cls = cls_c,
  sources.label.size = .6,
  ticks.label.size = .55)
```



```
#dev.off()
```

## Aquisição no último ano

```
FreqTab(dogs$aquisicao_u12)
```

	Category	Count	Proportion
1	nao	1027	0.96
2	sim	43	0.04

```
FreqTab(cats$aquisicao_u12)
```

	Category	Count	Proportion
1	nao	158	0.94
2	sim	10	0.06

## Destinos

```
sum(!is.na(dogs$destino)) / nrow(dogs)
```

```
[1] 0.1413738
```

```
sum(!is.na(cats$destino)) / nrow(cats)
```

```
[1] 0.1333333
```

```
FreqTab(dogs$destino)
```

	Category	Count	Proportion
1	morreu	124	0.701
2	doado	34	0.192
3	perdido	18	0.102



```
4 vendido      1      0.006
```

```
FreqTab(cats$destino)
```

	Category	Count	Proportion
1	morreu	15	0.577
2	doado	6	0.231
3	perdido	5	0.192

## Reposição

```
FreqTab(dogs$perdeu_animais)
```

	Category	Count	Proportion
1	nao	1070	0.855
2	sim	182	0.145

```
FreqTab(cats$perdeu_animais)
```

	Category	Count	Proportion
1	nao	168	0.862
2	sim	27	0.138

## Modelagem

### Condições iniciais e parâmetros

```
iasa_data <- GetDataIASA(dogs,
                          sex.col = "sexo",
                          female.label = "femea",
                          male.label = "macho",
                          sterilized.col = "castrado",
                          sterilized.label = "sim",
                          sterilized.ly.col = "castrado_u12",
                          sterilized.ly.label = "sim",
                          births.ly.col = "filhotes_u12",
                          species3.col = "especie3",
                          species.label = "cao",
                          sex3.col = "sexo3",
                          fate.col = "destino",
                          died.label = "morreu",
                          lost.label = "perdido",
                          acquisition.col = "aquisicao",
                          acquired.ly.col = "aquisicao_u12",
                          acquired.ly.label = "sim",
                          adopted.label = "adotou",
                          bought.label = "comprou",
                          acquisition.source.col = "municipio_de_aquisicao",
                          acquired.sterilized.col = "aquisicao_castrado",
                          destination.label = "pinhais",
                          total.estimate = 50443.79)
```

## Estimativas pontoais

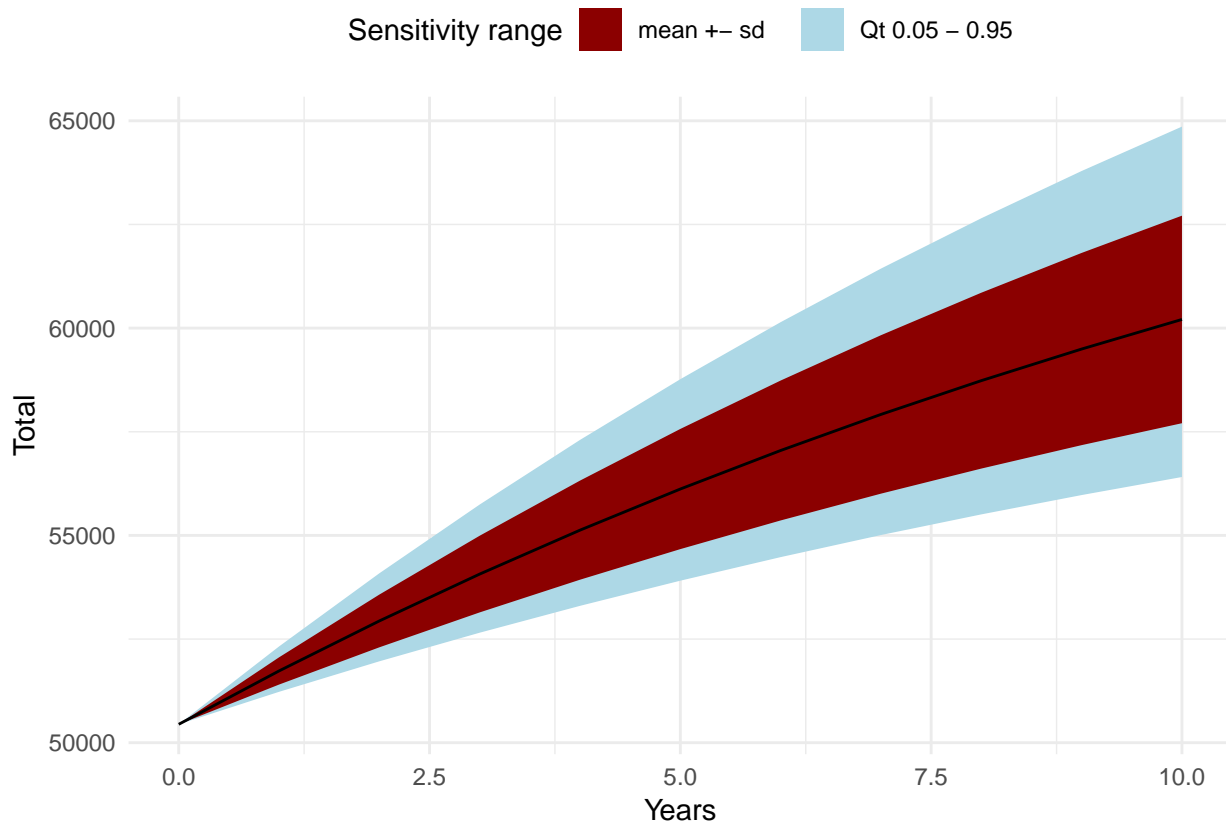
```
iasa <- SolveIASA(pars = iasa_data$pars, init = iasa_data$init, time = 0:10)
```

## Analise de sensibilidade

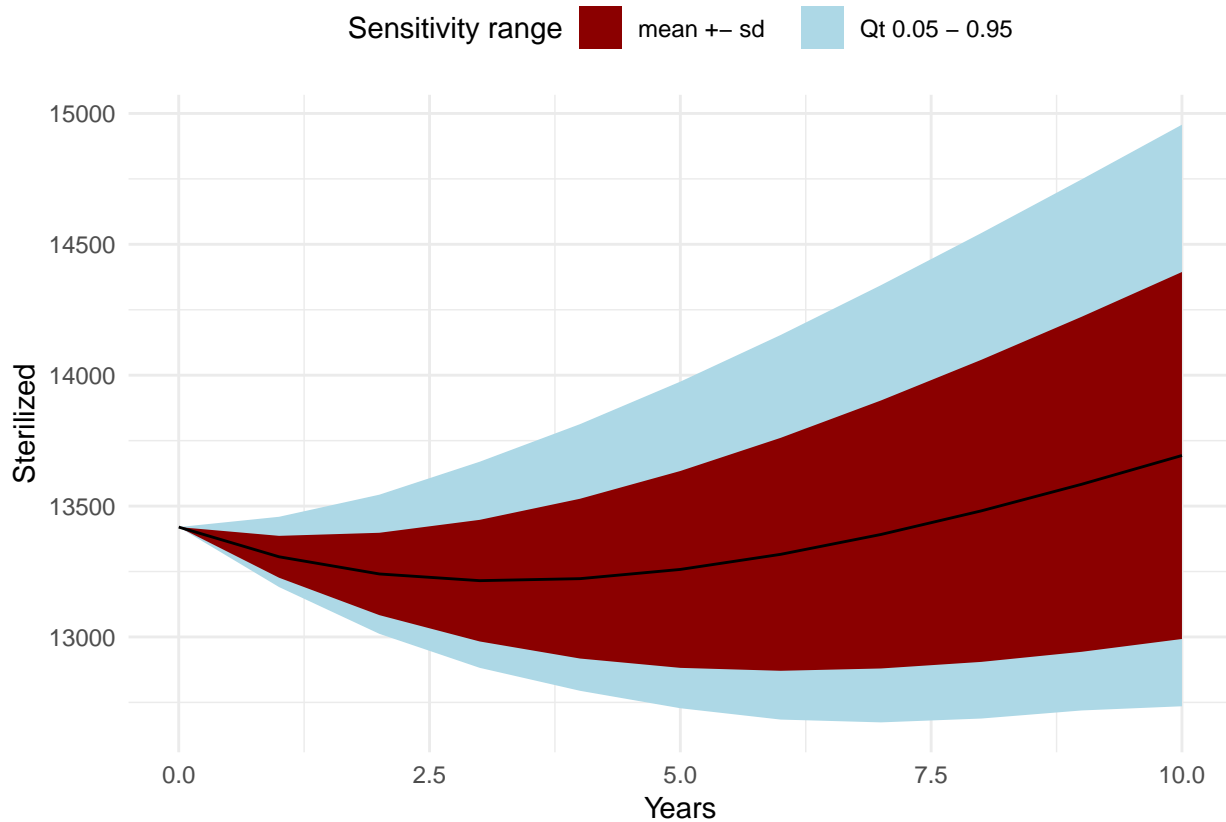
### Global

Os resultados podem mudar um pouco devido à estocasticidade. Quando comparados com os resultados do artigo, há uma variação maior se a versão do capm é 0.13.6 ou superior. As mudanças decorrentes da versão não alteram as conclusões do artigo e recomenda-se usar a versão mais recente.

```
ranges <- SetRanges(pars = iasa_data$pars, .1)
glob_N1 <- CalculateGlobalSens(iasa, ranges, sensv = "N1", all = TRUE)
glob_ns1 <- CalculateGlobalSens(iasa, ranges, sensv = "ns1", all = TRUE)
gg_glob_N1 <- PlotGlobalSens(glob_N1,
                             x.label = "Years",
                             y.label = "Total")
gg_glob_ns1 <- PlotGlobalSens(glob_ns1,
                              x.label = "Years",
                              y.label = "Sterilized")
gg_glob_N1
```



gg\_glob\_ns1



## Local

```
loc_N1 <- CalculateLocalSens(iasa, "N1")
loc_ns1 <- CalculateLocalSens(iasa, "ns1")

gg_loc_N1 <- PlotLocalSens(loc_N1)
gg_loc_ns1 <- PlotLocalSens(loc_ns1)

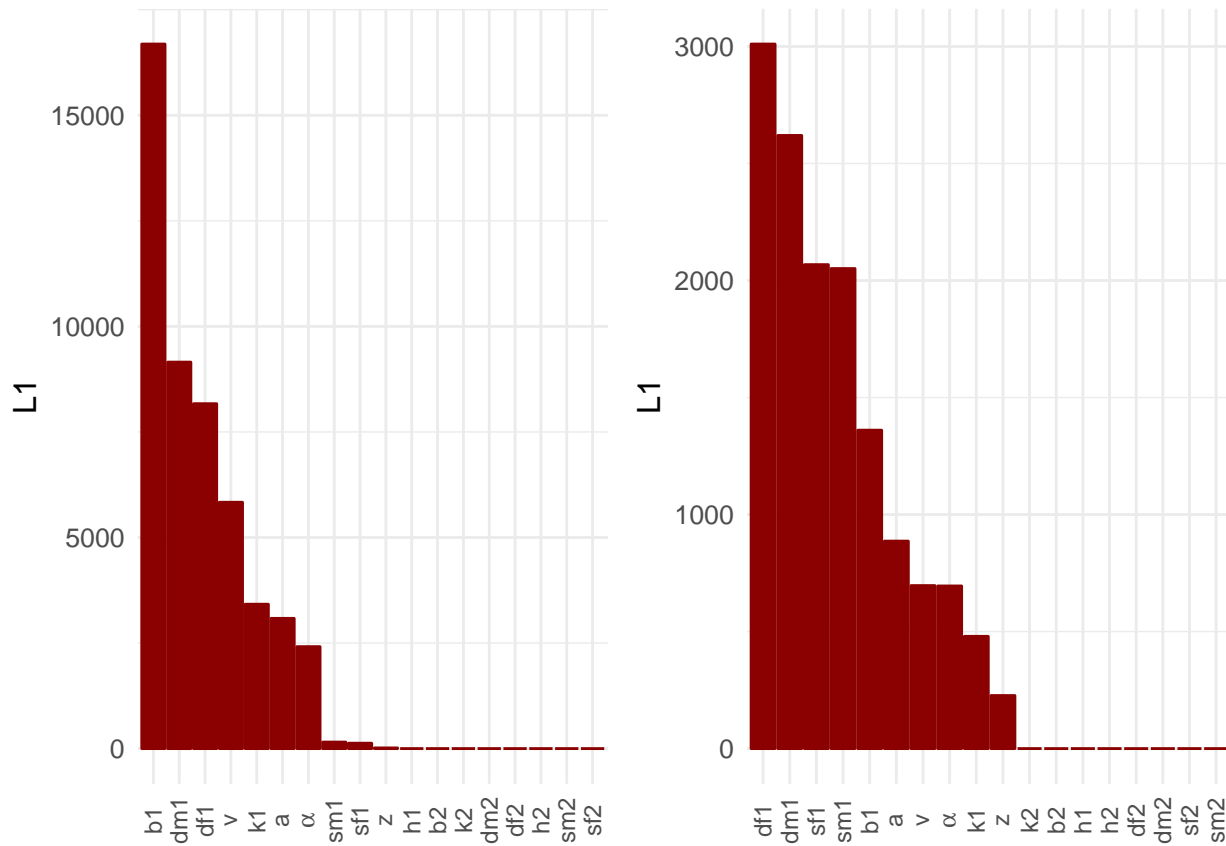
prop_L1_N1 <- round(summary(loc_N1)[, "L1"] /
                    max(summary(loc_N1)[, "L1"]) * 100, 2)
names(prop_L1_N1) <- rownames(summary(loc_N1))
sort(prop_L1_N1, decreasing = TRUE)
```

b1	dm1	df1	v	k1	a	alpha	sm1	sf1	z
100.00	54.86	48.96	34.96	20.49	18.48	14.47	0.92	0.76	0.09
b2	df2	dm2	sf2	sm2	k2	h1	h2		
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		

```
prop_L1_ns1 <- round(summary(loc_ns1)[, "L1"] /
                    max(summary(loc_ns1)[, "L1"]) * 100, 2)
names(prop_L1_ns1) <- rownames(summary(loc_ns1))
sort(prop_L1_ns1, decreasing = TRUE)
```

df1	dm1	sf1	sm1	b1	a	v	alpha	k1	z
100.00	87.03	68.69	68.14	45.19	29.45	23.13	23.08	15.93	7.53
b2	df2	dm2	sf2	sm2	k2	h1	h2		
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		

```
#tiff("./fig7.tiff", width = 4e3, height = 2e3, res = 300)
grid.arrange(gg_loc_N1, gg_loc_ns1, ncol = 2)
```



```
#dev.off()
```

## Mudança entre 2017 e 2027

Os resultados podem mudar um pouco devido à estocasticidade. Quando comparados com os resultados do artigo, há uma variação maior se a versão do capm é 0.13.6 ou superior. As mudanças decorrentes da versão não alteram as conclusões do artigo e recomenda-se usar a versão mais recente.

```
CalculatePopChange(glob_N1, variable = "N1", t2 = 10, t1 = 0)
```

```
[1] At t2, N1 will be 1.19 (1.12 - 1.29) times N1 at t1.
```

```
CalculatePopChange(glob_ns1, variable = "ns1", t2 = 10, t1 = 0)
```

```
[1] At t2, ns1 will be 1.02 (0.95 - 1.11) times ns1 at t1.
```

## População humana projetada (IBGE)

```
parana2017 <- 11320892
parana2027 <- 11929009
```

**Pinhais 2027**, pressupondo uma razão constante entre as populações de Pinhais e Paraná ao longo da década

```
pinhais2027 <- parana2027 * (pinhais2017 / parana2017)
```

### Mudança

```
pinhais2027 / pinhais2017
```

```
[1] 1.053716
```

```
pinhais2027 - pinhais2017
```

```
[1] 6953.313
```

### Taxas de esterilização propostas

Os resultados podem mudar um pouco devido à estocasticidade. Quando comparados com os resultados do artigo, há uma variação maior se a versão do capm é 0.13.6 ou superior. As mudanças decorrentes da versão não alteram as conclusões do artigo e recomenda-se usar a versão mais recente.

```
pars2 <- iasa_data$pars
pars2["sf1"] <- pars2["sf1"] * 1.3
pars2["sm1"] <- pars2["sm1"] * 1.3
iasa2 <- SolveIASA(pars = pars2, init = iasa_data$init, time = 0:10)
ranges2 <- SetRanges(pars = pars2, .1)
glob2_ns1 <- CalculateGlobalSens(iasa2, ranges2, sensv = "ns1", all = TRUE)
CalculatePopChange(glob2_ns1, variable = "ns1", t2 = 10, t1 = 0)
```

```
[1] At t2, ns1 will be 1.18 (1.09 - 1.28) times ns1 at t1.
```