

# Tamanho mínimo de amostra

Arquimedes Macedo. Tiago Rodrigues

## Contents

Objetivo . . . . .	2
Metodologia . . . . .	2
Resultados . . . . .	11
Sugestão de tamanho da amostra . . . . .	16

```
library(dplyr)
library(tidyr)
library(readxl)
library(knitr)
library(ggplot2)
library(ggthemes)
library(reshape2)
library(gridExtra)
library(vtable)
library(purrr)

# Centering figures chunk output
knitr::opts_chunk$set(out.height = "\\textheight", out.width = "\\textwidth",
                      out.extra = "keepaspectratio=true", fig.align = "center")

theme.base <- theme_minimal(base_size = 11) +
  theme(
    axis.text = element_text(size = 8),
    plot.title = element_text(hjust = 0.5, size = 12),
    axis.title = element_text(size = 10),
    panel.grid.major = element_line(colour = "grey90", linewidth = 0.5),
    panel.grid.minor = element_line(colour = adjustcolor("grey90", alpha.f = 0.5), linewidth = 0.25),
    panel.border = element_blank(),
    panel.background = element_blank(),
    plot.background = element_blank(),
    axis.line.x = element_line(colour = "grey"),
    axis.line.y = element_line(colour = "grey"),
  )

theme.no_legend <- theme(legend.position = "none")

theme.no_grid <- theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()
)

theme.no_axis <- theme(
```

```

axis.line.x = element_blank(),
axis.line.y = element_blank()
)

apply.theme.ts <- function() {
  list(
    scale_x_date(date_labels = "%b %d", date_breaks = "1 week"),
    theme.base + theme.no_legend +
      theme(
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank()
      )
  )
}

```

## Objetivo

Estimar a quantidade média de leads diários, com 80% de confiança, por anunciante, de anúncios de vendas de imóveis na cidade de Florianópolis (SC).

*Lead*: é um contato de um cliente em potencial que demonstrou interesse em um produto ou serviço.

IC de 80% foi escolhido devido à falta de informações (descrita logo mais), e também por ser este o valor máximo recomendado pela ABNT para avaliações de imóveis (NBR 14653).

## Metodologia

Almeja-se, a partir de uma lista de anúncios, realizar uma busca diária de leads, usando uma amostra dos anúncios, e, a partir destes dados, estimar a quantidade média de leads.

No entanto, entende-se que há limitações nas informações disponíveis, como:

- O número de leads por anúncio.
- Tempo total que o anúncio ficou ativo.
- A sazonalidade do mercado (oferta e demanda).
- A eficácia do anúncio (qualidade do anúncio, preço, localização, etc).
- A qualidade dos leads (interesse real ou apenas curiosidade).
- A distribuição subjacente dos leads ao longo do tempo.

Desta forma como um estudo piloto, foram obtidos leads diários, entre Janeiro e Julho de 2024, de anúncios de um anunciante na cidade alvo.

## Análise exploratória

O banco de dados é composto por 3 colunas:

- `id_registro`: identificador do lead.
- `data_criado_em`: dia que o lead foi gerado.
- `id_anuncio`: identificador do anúncio.

```

df_leads <- read_excel("dataset/leads.xlsx", col_types = c("numeric", "date", "text"))
df_leads$data_criado_em <- as.Date(df_leads$data_criado_em)
kable(head(df_leads))

```

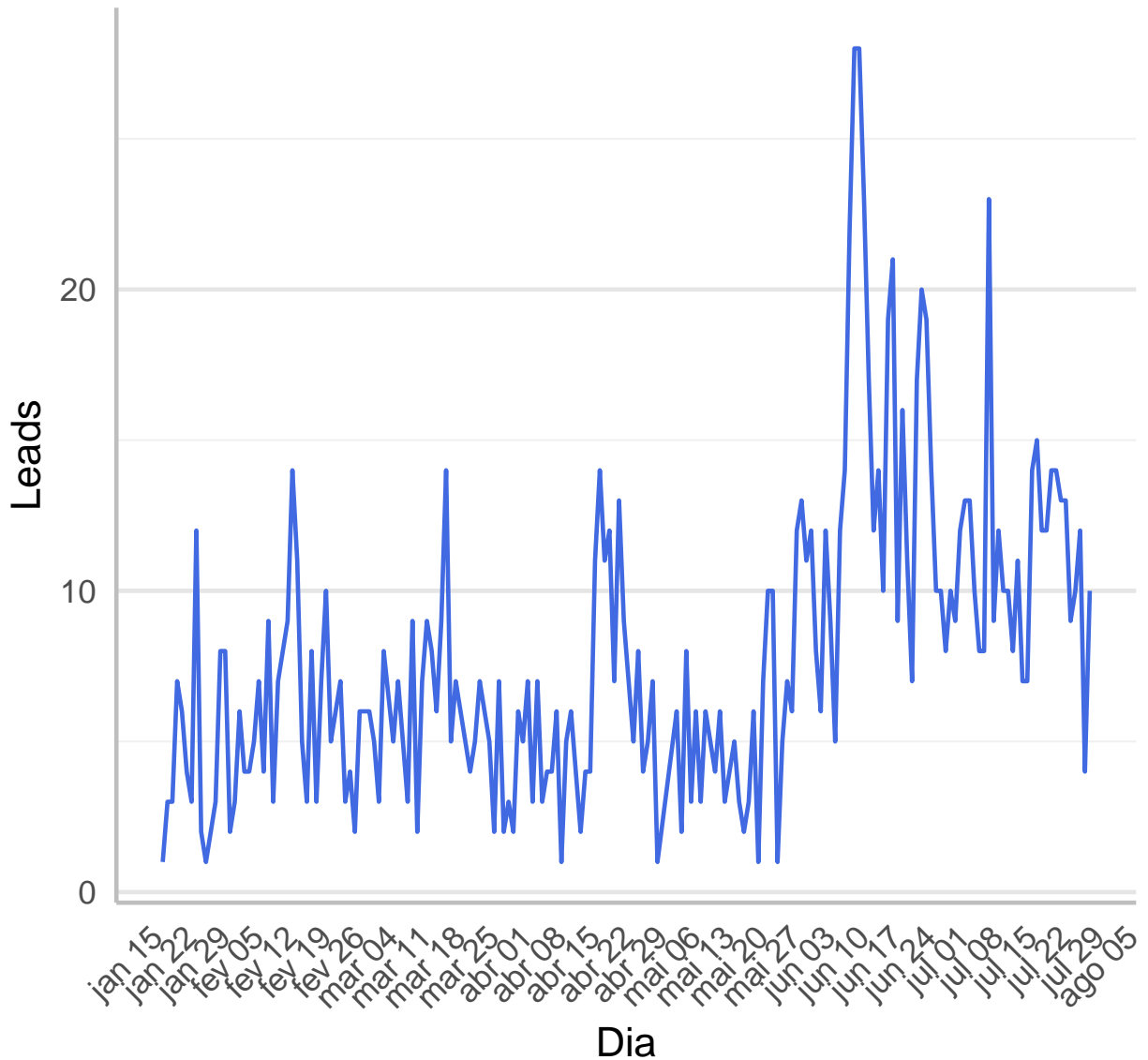
## Amostra dos dados

id_registro	data_criado_em	id_anuncio
1	2024-01-18	LRB3GK
2	2024-01-19	4I931S
3	2024-01-19	4WUWGH
4	2024-01-19	XNI94R
5	2024-01-20	HRDJQG
6	2024-01-20	CH8NIW

```
df_leads %>%
  group_by(data_criado_em) %>%
  summarise(leads = n()) %>%
  ggplot(aes(data_criado_em, leads)) +
  geom_line(color = "royalblue", linewidth = 0.5) +
  labs(title = "Leads diários",
       x = "Dia",
       y = "Leads") +
  apply.theme.ts()
```

Leads diários

## Leads diários



Notam-se picos em intervalos semi-regulares, o que pode indicar sazonalidade ou eventos específicos. Além disso, em Julho, houve uma alta variabilidade nos leads diários.

**Leads por anúncio** Vamos analisar a média diária de leads por anúncio.

Para isso, dividimos o número total de leads pelo número de anúncios únicos para cada dia.

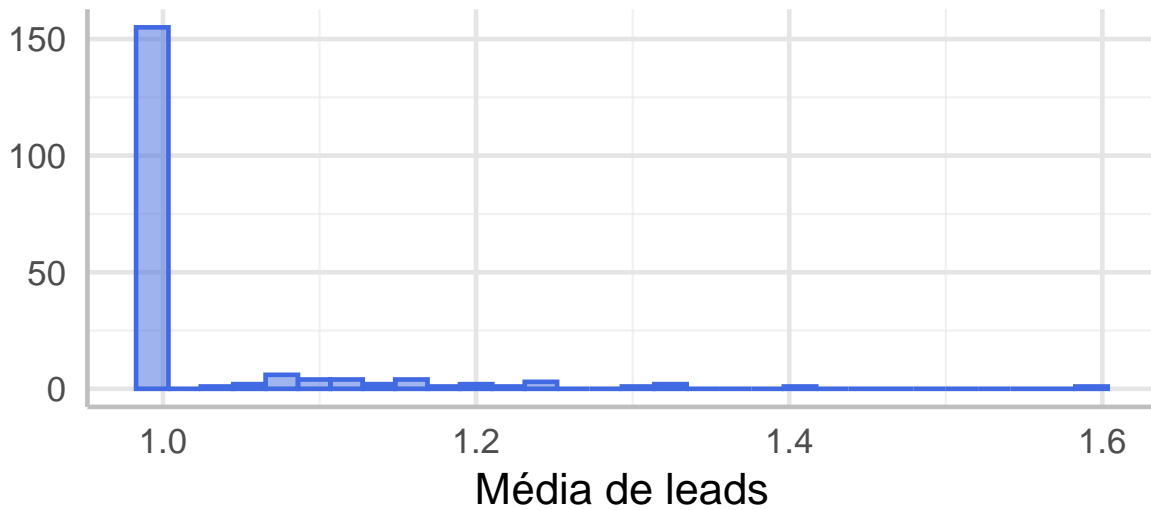
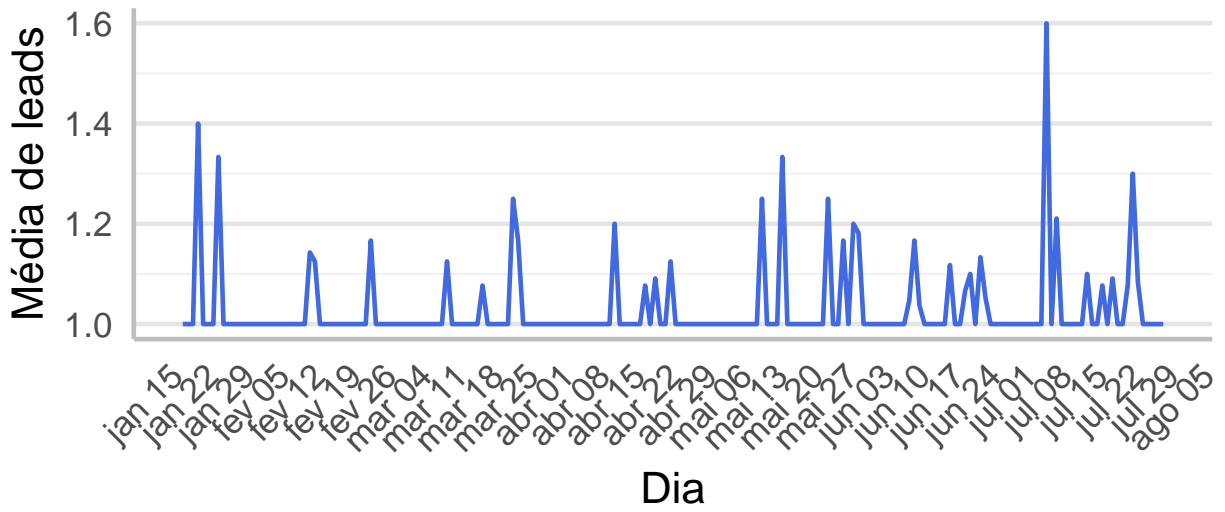
```
df_leads_incorrect_mean <- df_leads %>%  
  group_by(data_criado_em) %>%  
  summarise(mean = n()/length(unique(id_anuncio)))  
  
grid.arrange(  
  df_leads_incorrect_mean %>%  
    ggplot(aes(data_criado_em, mean)) +
```

```

    geom_line(color = "royalblue", linewidth = 0.5) +
    labs(title = "Média de leads por dia",
         x = "Dia",
         y = "Média de leads") +
    apply.theme.ts(),
df_leads_incorrect_mean %>%
  ggplot(aes(mean)) +
  geom_histogram(bins = 30, color = "royalblue", fill = "royalblue", alpha = 0.5) +
  labs(title = "",
       x = "Média de leads",
       y = "") +
  theme.base + theme.no_legend,
nrow = 2
)

```

## Média de leads por dia



Será que é só isso mesmo?



### Estimativa da média

Claro que não! A média diária de leads por anúncio é uma estimativa incorreta, pois não considera a quantidade de anúncios ativos em cada dia, e acaba gerando um viés. Para corrigir isso, vamos completar os dados com zeros para os dias sem leads.

Isto é, vamos pegar o primeiro e o último dia que o anúncio teve leads, e criar novos registros entre estas datas, para dias sem lead.

```
df_leads_complete <- df_leads %>%
  group_by(id_anuncio, data_criado_em) %>%
  summarise(leads = n(), .groups = 'drop') %>%
  group_by(id_anuncio) %>%
  # Creates a list of dataframes by id
  tidyr::nest() %>%
  mutate(
    # Creates a sequence of dates by id
    date_seq = map(data, ~seq(min(.$data_criado_em), max(.$data_criado_em), by = "day")),
    # Completes the missing dates
    data = map2(
      data, date_seq,
      \(data_, seq_) {
        data_ %>%
          complete(data_criado_em = seq_, fill = list(leads = 0))
      }
    )
  ) %>%
  # Removes the auxiliary column
  select(-date_seq) %>%
  # Unnests the data
  unnest(data)

kable(head(df_leads_complete))
```

id_anuncio	data_criado_em	leads
00OPP2	2024-03-10	1
00SLR7	2024-06-29	1
00TPRF	2024-06-23	1
02NTL4	2024-04-20	1
02NTL4	2024-04-21	0
02NTL4	2024-04-22	0

A partir desta correção, temos as seguintes médias diárias.

```
df_leads_daily <- df_leads_complete %>%
  group_by(data_criado_em) %>%
  summarise(mean = mean(leads),
            total_leads = sum(leads),
            active_listings = n_distinct(id_anuncio))

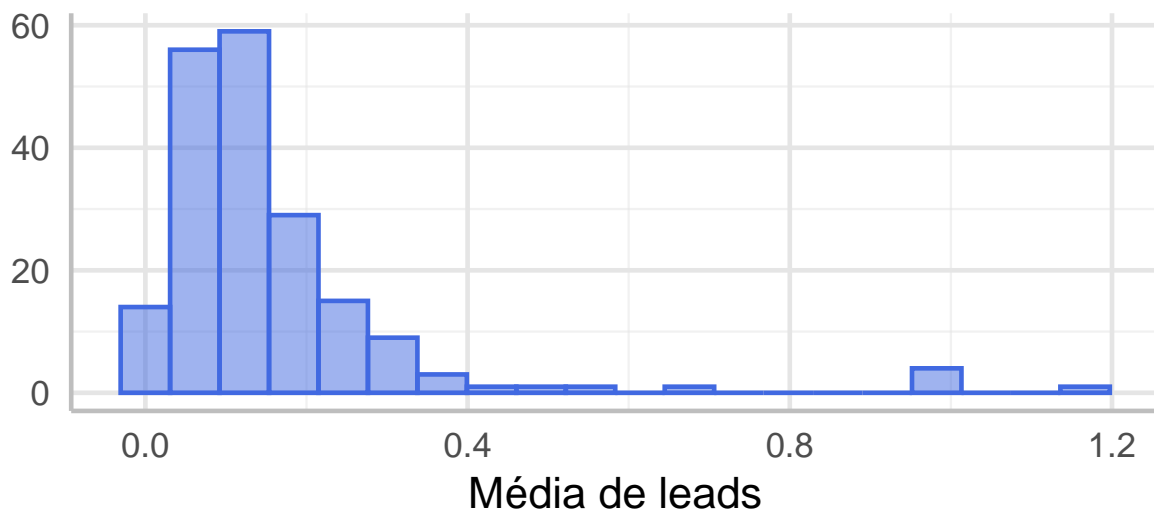
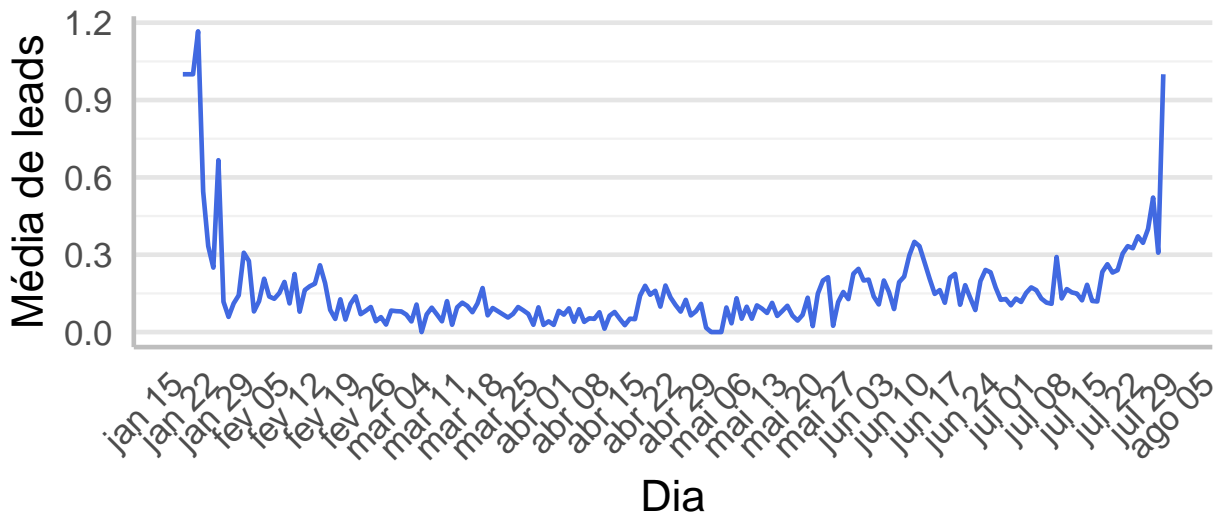
grid.arrange(
  df_leads_daily %>%
    ggplot(aes(data_criado_em, mean)) +
    geom_line(color = "royalblue", linewidth = 0.5) +
    labs(title = "Média de leads por dia",
         x = "Dia",
         y = "Média de leads") +
  apply.theme.ts(),
  df_leads_daily %>%
    ggplot(aes(mean)) +
```

```

geom_histogram(bins = 20, color = "royalblue", fill = "royalblue", alpha = 0.5) +
labs(title = "",
      x = "Média de leads",
      y = "") +
theme.base + theme.no_legend,
nrow = 2
)

```

## Média de leads por dia



Mas não está totalmente correto...



Table 3: Registros corrigidos

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 50	Pctl. 75	Max
leads	11253	0.12	0.34	0	0	0	0	4



Lembrando que esta é uma aproximação e não corresponde totalmente ao que de fato aconteceu, para computar a verdadeira média, precisaríamos da listagem de todos os anúncios ativos no dia.

Nota-se, também, que existem pontos extremos no início e no fim da série, isso pode ser explicado por anúncios que estavam ativos antes do início do período analisado ou que apareceram um pouco antes do fim.

Portanto vamos analisar apenas entre 01/02/2024 e 20/07/2024.

```
df_leads_complete_filtered <- df_leads_complete %>%
  filter(between(data_criado_em, as.Date("2024-02-01"), as.Date("2024-07-20")))
```

```
df_leads_daily_filtered <- df_leads_complete_filtered %>%
  group_by(data_criado_em) %>%
  summarise(mean = mean(leads),
            total_leads = sum(leads),
            active_listings = n_distinct(id_anuncio))
```

```
sumtable(df_leads_complete_filtered, add.median = T, title = "Registros corrigidos")
```

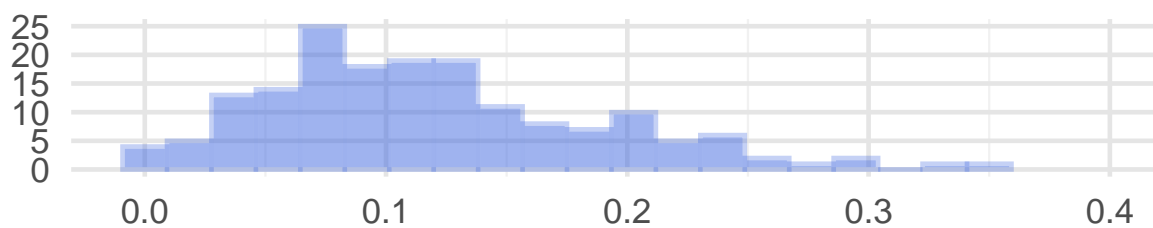
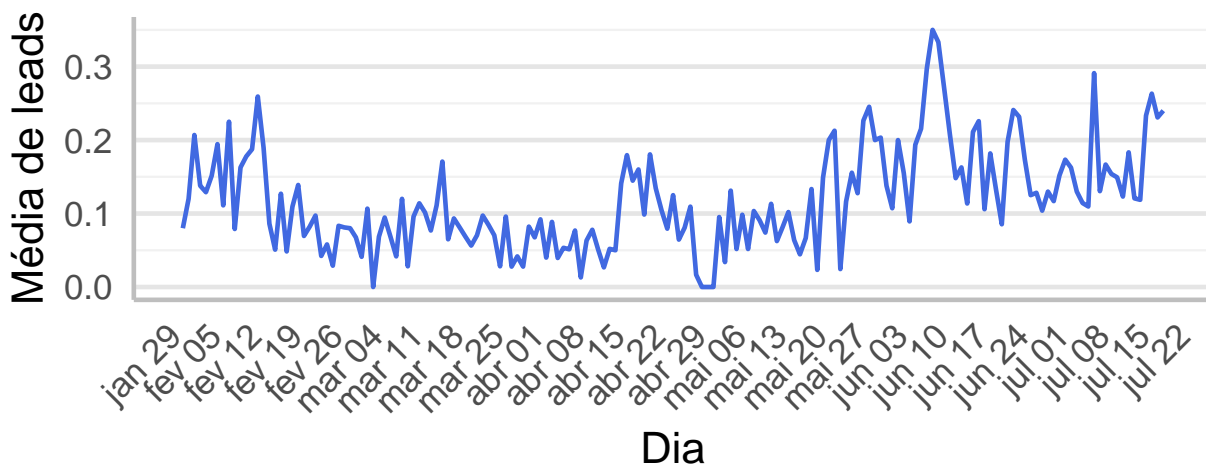
```
grid.arrange(
  df_leads_daily_filtered %>%
    ggplot(aes(data_criado_em, mean)) +
    geom_line(color = "royalblue", linewidth = 0.5) +
    labs(title = "Média de leads por dia",
         x = "Dia",
         y = "Média de leads") +
    apply.theme.ts(),
  df_leads_daily_filtered %>%
    ggplot(aes(mean)) +
    coord_cartesian(xlim = c(-0.01, 0.4)) +
    geom_histogram(bins = 20, color = adjustcolor("royalblue", alpha.f = 0.3), fill = "royalblue", alpha.f = 0.3) +
    labs(title = "",
         x = "",
         y = "") +
    theme.base + theme.no_legend + theme.no_axis +
    theme(panel.grid.minor.y = element_blank()),
  df_leads_daily_filtered %>%
    ggplot(aes(mean)) +
    coord_cartesian(xlim = c(-0.02, 0.4)) +
    geom_boxplot(color = adjustcolor("royalblue", alpha.f = 0.8), fill = "royalblue", alpha = 0.5) +
    labs(title = "",
         x = "",
```

```

y = """) +
theme.base + theme.no_legend + theme.no_axis +
theme(axis.text.y = element_blank(),
      axis.ticks.y = element_blank(),
      panel.grid.major.y = element_blank(),
      panel.grid.minor.y = element_blank()),
nrow = 3,
heights = c(3, 2, 1.5)
)

```

## Média de leads por dia



```

leads_mean <- mean(df_leads_complete_filtered$leads)
leads_sd <- sd(df_leads_complete_filtered$leads)

```

Table 4: Média de leads por dia

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 50	Pctl. 75	Max
mean	171	0.12	0.069	0	0.068	0.11	0.16	0.35
total_leads	171	7.8	5.1	0	4	7	10	28
active_listings	171	66	14	25	58	71	76	93

```
sumtable(df_leads_daily_filtered, add.median = T, title = "Média de leads por dia")
```

## Resultados

Assim, apesar dos pesares, temos uma média de ~0.12 leads por dia, com um desvio padrão de ~0.337.

### Tamanho da amostra

Calculamos que o tamanho da amostra, a partir da equação

$$n' \geq \left( Z_{\alpha/2} \frac{\sigma}{e} \right)^2$$

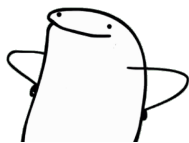
$$n = n' \cdot \frac{N - n}{N - 1}$$

onde,  $Z_{\alpha/2}$  é o valor crítico da distribuição normal, e é a margem de erro,  $\sigma$  é o desvio padrão,  $N$  é o tamanho da população,  $n'$  é o tamanho da amostra com amostra aleatória simples com reposição (AASc), e,  $n$  é o tamanho da amostra sem reposição.

```
confidence <- 0.8
z <- abs(qnorm((1 - confidence) / 2))
e <- 0.1
sigma <- leads_sd
size_population <- nrow(df_leads_complete_filtered)

computed_sample_size_aasc <- (z * sigma / e)^2
computed_sample_size <- ceiling(
  computed_sample_size_aasc *
  (size_population - computed_sample_size_aasc) / (size_population - 1)
)
```

**Ta-dá!** Para obter uma margem de erro de 0.1 com 80% de confiança, precisamos de uma amostra de 19 anúncios.



### Cálculo amostral

```
nivel_conf <- 0.80
z <- qnorm(1 - (1 - nivel_conf) / 2)
```

```
n_calculado <- ceiling(z^2 * (var(df_leads_daily_filtered$mean)/0.05^2))
print(paste("n calculado:", n_calculado))
```

Tamanho da amostra: n\_calculado

```
## [1] "n calculado: 4"
```

```
margem_erro <- z * (sd(df_leads_daily_filtered$mean) / sqrt(n_calculado))
print(paste("margem de erro:", margem_erro))
```

Margem de erro calculada: margem\_erro

```
## [1] "margem de erro: 0.0441555477843345"
```

```
num_amostras <- 1000

set.seed(2024) # Definida uma semente
values_mean_amostral <- replicate(num_amostras, {
  amostra <- sample(df_leads_daily_filtered$mean, n_calculado, replace = FALSE)
  mean(amostra)
})
print(values_mean_amostral)
```

Geração de 1000 amostras aleatórias das médias diárias: values\_mean\_amostral

```
## [1] 0.11501045 0.12053098 0.11109110 0.10297871 0.12457675 0.08301267
## [7] 0.13621481 0.09576870 0.17370370 0.05480625 0.15076811 0.11822179
## [13] 0.06973701 0.21897062 0.11221218 0.12055298 0.09572781 0.14682857
## [19] 0.09995705 0.05851051 0.12456044 0.09026366 0.17660294 0.09737654
## [25] 0.09026039 0.09151259 0.08406773 0.15210757 0.11549145 0.13209376
## [31] 0.18579282 0.10720585 0.08188196 0.17375969 0.16037594 0.08381849
## [37] 0.18920420 0.09575427 0.14148180 0.11331197 0.05615262 0.14479614
## [43] 0.14839135 0.05439013 0.14710229 0.04464182 0.10130275 0.15582309
## [49] 0.16037055 0.13612197 0.10857065 0.19779777 0.10495531 0.08819991
## [55] 0.13716130 0.11548992 0.12026925 0.08177614 0.12539557 0.09690003
## [61] 0.06253743 0.13316670 0.10437695 0.16655359 0.09011701 0.16318125
## [67] 0.07541667 0.09416823 0.07755831 0.09966420 0.21141645 0.12409310
## [73] 0.13459221 0.09739927 0.07819444 0.22261667 0.14709124 0.09249035
## [79] 0.14274059 0.14664408 0.10412306 0.12875817 0.12683412 0.15698306
## [85] 0.13361571 0.14767070 0.10691453 0.18010415 0.07505994 0.07365862
## [91] 0.05563862 0.15492588 0.10071009 0.12231382 0.18307334 0.16796136
## [97] 0.16924225 0.11252412 0.13741518 0.10899131 0.05171785 0.12459350
## [103] 0.11973539 0.14250284 0.09655303 0.19795997 0.13285505 0.10289261
## [109] 0.08187977 0.15841841 0.11650789 0.17134175 0.08461952 0.08333333
## [115] 0.07988202 0.10219928 0.11430107 0.14200924 0.09865689 0.10524944
## [121] 0.11056171 0.13017590 0.13263823 0.09266368 0.13745556 0.13041105
## [127] 0.13313928 0.15645058 0.08257447 0.11536797 0.09000323 0.12441300
## [133] 0.10834608 0.10457235 0.15706415 0.09617105 0.13937387 0.12318490
## [139] 0.15682265 0.09489734 0.11882562 0.09156508 0.09849630 0.15600768
## [145] 0.09624745 0.08085345 0.17345467 0.09683952 0.11124562 0.14173692
## [151] 0.11697149 0.12240599 0.09277280 0.13961578 0.10461560 0.09075611
## [157] 0.09492356 0.17311064 0.06788427 0.09037399 0.09075946 0.18313253
## [163] 0.14648089 0.11314233 0.11127471 0.12441335 0.10963490 0.15865297
## [169] 0.13987781 0.09816841 0.09066198 0.14879102 0.11585039 0.14239148
```

```

## [175] 0.12675835 0.15667129 0.12034901 0.07786546 0.10691743 0.17233643
## [181] 0.09082228 0.05651069 0.11724354 0.15231536 0.15344379 0.14997163
## [187] 0.13310505 0.16690780 0.16489638 0.13795081 0.15371224 0.13013116
## [193] 0.06132624 0.16508621 0.13232323 0.09575545 0.13001565 0.14185455
## [199] 0.07380289 0.06417078 0.11956225 0.15088722 0.16705912 0.14473253
## [205] 0.11677991 0.18878441 0.08046140 0.19508785 0.07719941 0.13261991
## [211] 0.12192550 0.14404078 0.16157088 0.10997336 0.13750038 0.08474427
## [217] 0.09083762 0.19018290 0.08398325 0.14303418 0.14507659 0.14932372
## [223] 0.07750367 0.18224437 0.10746407 0.12851424 0.10737435 0.16592290
## [229] 0.12264137 0.11156851 0.12006108 0.12942590 0.08645317 0.13015900
## [235] 0.08879254 0.11896437 0.11273119 0.10369417 0.11182439 0.12057211
## [241] 0.10755890 0.12051264 0.12023252 0.15963686 0.13971196 0.15125487
## [247] 0.08972557 0.08070175 0.12901310 0.10800067 0.13612237 0.08487927
## [253] 0.13215812 0.09498067 0.10086874 0.12373715 0.16255768 0.09511816
## [259] 0.16363940 0.14820529 0.13481801 0.06143332 0.14618352 0.14005666
## [265] 0.13562500 0.10142517 0.10441521 0.14038542 0.15070654 0.14718663
## [271] 0.09490923 0.08166202 0.11766585 0.10852093 0.13195011 0.06328730
## [277] 0.16666748 0.14602155 0.16296794 0.08755647 0.12424623 0.07444876
## [283] 0.18840290 0.14744230 0.12193028 0.06329427 0.09174012 0.10316492
## [289] 0.17066579 0.12185497 0.09763171 0.06340713 0.09680062 0.18242012
## [295] 0.09534100 0.14019413 0.10561469 0.14673765 0.13266720 0.18131070
## [301] 0.09422644 0.11325989 0.16053879 0.13781073 0.12500364 0.06608991
## [307] 0.12181467 0.11600862 0.10805478 0.10168172 0.12492159 0.11600112
## [313] 0.07551317 0.06502223 0.09110823 0.14861284 0.09409332 0.08446886
## [319] 0.15052464 0.11438307 0.11164044 0.13806324 0.15724105 0.08754098
## [325] 0.10921573 0.09541464 0.09030974 0.07268261 0.12670455 0.15101090
## [331] 0.08801803 0.14760530 0.16776896 0.15537187 0.13351153 0.16944662
## [337] 0.07325878 0.15153017 0.12656988 0.12716106 0.09108394 0.10767123
## [343] 0.16199713 0.09255574 0.09576167 0.15394901 0.09075741 0.21564013
## [349] 0.12001959 0.08168803 0.06384389 0.14140620 0.08022774 0.06444327
## [355] 0.10287677 0.07461527 0.09864820 0.15056130 0.06143856 0.17319352
## [361] 0.07398140 0.08839333 0.18788972 0.09120520 0.13487847 0.07534088
## [367] 0.06312955 0.12705378 0.11028139 0.07062420 0.12800726 0.07826742
## [373] 0.16466256 0.06018227 0.08882037 0.15357952 0.11770299 0.20127650
## [379] 0.09568204 0.07494172 0.10467095 0.13306524 0.11785317 0.12359634
## [385] 0.18304318 0.14135985 0.17651738 0.10706933 0.13154745 0.19612937
## [391] 0.11137750 0.07828773 0.11560231 0.15437383 0.09643543 0.07669388
## [397] 0.07369041 0.14024758 0.18444767 0.08064707 0.08952425 0.16353727
## [403] 0.09845417 0.09351054 0.15303017 0.06666223 0.13253221 0.10224385
## [409] 0.09491994 0.12094928 0.14838051 0.10516654 0.12866701 0.06028531
## [415] 0.07571568 0.09414094 0.09963901 0.15265755 0.11413845 0.14633291
## [421] 0.08928612 0.10712758 0.10956925 0.08517599 0.11509770 0.11686368
## [427] 0.11803050 0.15401210 0.13935227 0.07793311 0.12998391 0.21403708
## [433] 0.14637662 0.11751408 0.08850310 0.10934700 0.10652078 0.13275141
## [439] 0.12726447 0.16197786 0.14539928 0.14369534 0.07195517 0.14513180
## [445] 0.16041293 0.12487871 0.09424433 0.10290916 0.12458333 0.14246529
## [451] 0.09116191 0.06619709 0.13165001 0.06580951 0.05959206 0.09439066
## [457] 0.10431267 0.09549315 0.14546203 0.15633667 0.15551851 0.06599261
## [463] 0.13834207 0.09745005 0.11628013 0.13204225 0.09065799 0.11164104
## [469] 0.19199029 0.10760784 0.16878900 0.11065949 0.07152330 0.10753159
## [475] 0.19462438 0.15285131 0.11219450 0.16493657 0.17159894 0.09994690
## [481] 0.12420271 0.10314597 0.15541167 0.15784660 0.05433068 0.10642019
## [487] 0.12885383 0.09672592 0.14524544 0.12337382 0.12606127 0.09668833
## [493] 0.15114182 0.04237041 0.13549306 0.09411039 0.18726221 0.13444767

```

## [499] 0.11743742 0.16192968 0.11376408 0.11933795 0.09629542 0.12896586  
 ## [505] 0.12219267 0.14217301 0.11489622 0.17512582 0.11685685 0.08552226  
 ## [511] 0.20249704 0.10588134 0.10449846 0.10213375 0.16693913 0.17752208  
 ## [517] 0.08568489 0.11397715 0.07364404 0.08783926 0.11869374 0.06462019  
 ## [523] 0.12151683 0.13323876 0.13834767 0.08520906 0.05902762 0.09064746  
 ## [529] 0.10359856 0.12251711 0.10352958 0.13130342 0.12640096 0.13483767  
 ## [535] 0.15160256 0.16458560 0.16276178 0.14242274 0.07653356 0.12497071  
 ## [541] 0.12991099 0.11283061 0.06321590 0.11406861 0.15163692 0.11470624  
 ## [547] 0.10718524 0.05871578 0.15156532 0.07709132 0.15397564 0.13153010  
 ## [553] 0.11565936 0.11040206 0.07669039 0.11068129 0.18954988 0.10252352  
 ## [559] 0.11154099 0.05782496 0.08524272 0.10593021 0.18473127 0.09471143  
 ## [565] 0.10435739 0.13582725 0.07702156 0.17527541 0.11679022 0.10984573  
 ## [571] 0.12067373 0.06744201 0.11875618 0.09728205 0.12798122 0.12730242  
 ## [577] 0.16965029 0.13288345 0.09112676 0.08762969 0.16085706 0.15357972  
 ## [583] 0.08802810 0.12121828 0.15746212 0.09572978 0.20961277 0.11052632  
 ## [589] 0.13497380 0.11259923 0.08522472 0.11273006 0.12462760 0.18238880  
 ## [595] 0.10750000 0.14478218 0.08144231 0.14484548 0.10905679 0.14349399  
 ## [601] 0.09389817 0.19483871 0.12862008 0.11205189 0.10506490 0.14397556  
 ## [607] 0.07611922 0.18773228 0.08389578 0.12023705 0.11112773 0.13134034  
 ## [613] 0.07239113 0.16736180 0.13345738 0.10314285 0.13272500 0.11416817  
 ## [619] 0.08257859 0.08978187 0.09032308 0.14382845 0.13655819 0.12243121  
 ## [625] 0.13307020 0.09884796 0.13197090 0.11699630 0.08393689 0.11234698  
 ## [631] 0.08320202 0.09739503 0.08851751 0.13042771 0.13424974 0.10636523  
 ## [637] 0.10896307 0.09434831 0.10737703 0.09841115 0.12072168 0.05854226  
 ## [643] 0.12133313 0.13698718 0.17621255 0.10181923 0.08299623 0.16296748  
 ## [649] 0.18918089 0.15682856 0.11544827 0.14323309 0.21024887 0.11457024  
 ## [655] 0.11639675 0.10411297 0.09346921 0.08876689 0.12969704 0.09921662  
 ## [661] 0.13106053 0.08763889 0.10833333 0.12685364 0.13740153 0.15813783  
 ## [667] 0.11921332 0.03650976 0.12774600 0.10686078 0.12261613 0.14403362  
 ## [673] 0.11256260 0.15952336 0.11876910 0.10632060 0.10300039 0.10713472  
 ## [679] 0.09885644 0.11552254 0.15342163 0.14623273 0.14755866 0.11297365  
 ## [685] 0.10210409 0.14046654 0.05859516 0.10834175 0.07750883 0.11015389  
 ## [691] 0.14006763 0.11351659 0.13531130 0.16961175 0.16557118 0.15861565  
 ## [697] 0.07440031 0.11826754 0.12473361 0.11327543 0.13320431 0.07247569  
 ## [703] 0.10741490 0.07639196 0.11170625 0.14370208 0.15531478 0.18051708  
 ## [709] 0.15342593 0.11525129 0.13023596 0.08626489 0.09572467 0.12201919  
 ## [715] 0.11810581 0.10947978 0.08460341 0.06779892 0.09362751 0.19387042  
 ## [721] 0.10944277 0.13986545 0.13345868 0.12800300 0.08983650 0.11042604  
 ## [727] 0.06485405 0.11801663 0.18409136 0.05220509 0.15963701 0.15706503  
 ## [733] 0.07190155 0.12243411 0.14874100 0.08541763 0.12105263 0.13378995  
 ## [739] 0.08721195 0.08465808 0.12305963 0.03665612 0.13650677 0.10855024  
 ## [745] 0.11588731 0.07204548 0.10455856 0.08623450 0.19206615 0.11555556  
 ## [751] 0.10817166 0.08591270 0.11328157 0.12033419 0.16419774 0.09963933  
 ## [757] 0.07278327 0.08672099 0.11936407 0.18561624 0.09658862 0.15492693  
 ## [763] 0.11783642 0.09436067 0.15880127 0.11734139 0.11607920 0.12083572  
 ## [769] 0.13464093 0.07321319 0.06897396 0.13167497 0.12261674 0.18598808  
 ## [775] 0.08766192 0.14520982 0.14188312 0.11129184 0.10070849 0.13973793  
 ## [781] 0.14473683 0.04293567 0.16352644 0.12053357 0.13136324 0.09109015  
 ## [787] 0.14949309 0.15028853 0.07275531 0.10873192 0.10353020 0.15067862  
 ## [793] 0.13295997 0.09903794 0.12126367 0.10619642 0.11577551 0.09741416  
 ## [799] 0.10303083 0.16336209 0.07468448 0.12688672 0.20067718 0.09438001  
 ## [805] 0.13411645 0.09800125 0.12003978 0.11924435 0.16617063 0.10460820  
 ## [811] 0.07899351 0.13074128 0.19100325 0.12034989 0.15406506 0.11311666  
 ## [817] 0.10948276 0.17641148 0.13147416 0.07137629 0.07157264 0.13987170

```
## [823] 0.11817033 0.07022862 0.09065518 0.08655112 0.11153726 0.09377354
## [829] 0.15348238 0.14541270 0.11279333 0.08029902 0.10988914 0.09864748
## [835] 0.04908732 0.12400598 0.11483336 0.07111494 0.13699660 0.15038246
## [841] 0.06788227 0.10735348 0.04728320 0.11848263 0.16274482 0.19712644
## [847] 0.15891332 0.15566432 0.14326144 0.07982504 0.09615245 0.20403481
## [853] 0.15933310 0.15677834 0.10596646 0.13324786 0.10344433 0.10080897
## [859] 0.11032091 0.13895458 0.15151250 0.13212747 0.09568897 0.11958796
## [865] 0.09232497 0.17784743 0.10350735 0.15174085 0.13357947 0.11253725
## [871] 0.11099545 0.15021380 0.12243308 0.07417115 0.17341620 0.13148381
## [877] 0.15066770 0.14064624 0.11848404 0.11375182 0.11698614 0.11072613
## [883] 0.17195626 0.10726689 0.07839983 0.11812615 0.08122367 0.07946451
## [889] 0.11110739 0.13470548 0.10818630 0.09087566 0.09871795 0.15650427
## [895] 0.09196237 0.12924976 0.10883028 0.07098787 0.14520952 0.17086172
## [901] 0.07636307 0.12141721 0.07170370 0.18029423 0.06057971 0.14455351
## [907] 0.15500743 0.09768901 0.10056610 0.16145421 0.14629122 0.17222244
## [913] 0.15901300 0.07292711 0.13571886 0.14330383 0.09477174 0.17895336
## [919] 0.13061900 0.11766351 0.09324853 0.17211165 0.14759512 0.10463040
## [925] 0.12562443 0.09895584 0.10279001 0.09292364 0.13075414 0.14173815
## [931] 0.18028221 0.19474099 0.13085720 0.15646831 0.12495450 0.12288387
## [937] 0.12247725 0.11607622 0.14358858 0.11321646 0.08727453 0.08809829
## [943] 0.14305634 0.12203796 0.16651556 0.10043986 0.14905618 0.15694084
## [949] 0.10592109 0.09173419 0.10444314 0.11203691 0.11441911 0.05546775
## [955] 0.06946883 0.11278551 0.16946171 0.08301211 0.09921769 0.15009778
## [961] 0.14001390 0.11035546 0.17359703 0.11968432 0.11328921 0.16894285
## [967] 0.08287174 0.14719281 0.12183009 0.10592395 0.11273211 0.09451588
## [973] 0.07743228 0.11645019 0.12862903 0.12802290 0.11093449 0.09666231
## [979] 0.12612719 0.16818479 0.09892972 0.18486479 0.08561368 0.16581661
## [985] 0.11188554 0.12999445 0.12659004 0.07324589 0.14778935 0.14147065
## [991] 0.13944064 0.07943051 0.11850916 0.14996738 0.08294417 0.14984653
## [997] 0.13354798 0.16512213 0.11628404 0.06127845
```

```
# Média das 1000 amostras geradas de tamanho n_calculado para estimar a média populacional
```

```
mean_amostrai <- mean(values_mean_amostrai)
print(paste("Média amostral:", mean_amostrai))
```

Média amostral das 1000 amostras obtidas: mean\_\_amostral

```
## [1] "Média amostral: 0.119999608259177"
```

```
# Calculando o maior e o menor valor das médias amostrais
```

```
highest_media <- max(values_mean_amostrai)
lowest_media <- min(values_mean_amostrai)
```

```
# Calculando o erro (diferença entre o maior e o menor valor)
```

```
erro <- highest_media - lowest_media
```

```
# Exibindo os resultados
```

```
print(paste("Maior média amostral:", highest_media))
```

Cálculo do erro das médias amostrais obtidas: erro

```
## [1] "Maior média amostral: 0.222616666012892"
```

```
print(paste("Menor média amostral:", lowest_media))

## [1] "Menor média amostral: 0.0365097630920416"

print(paste("Erro (diferença entre maior e menor média):", erro))

## [1] "Erro (diferença entre maior e menor média): 0.186106902920851"
```

## Sugestão de tamanho da amostra

Para trabalhos futuros sugerimos que o tamanho da amostra seja de