Tamanho mínimo de amostra

Arquimedes Macedo. Tiago Rodrigues

Contents


```
library(dplyr)
library(tidyr)
library(readxl)
library(knitr)
library(ggplot2)
library(ggridges)
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 = 10, face = "bold"),
    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()
)

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


# Theme for timeseries
apply.theme.ts <- function() {
  list(
    apply.theme.ts.legend(),
    theme.no_legend
  )
}
```

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).

Com um erro máximo de 0.05, usando Amostragem Aleatório Simples sem Reposição (AASs).

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.

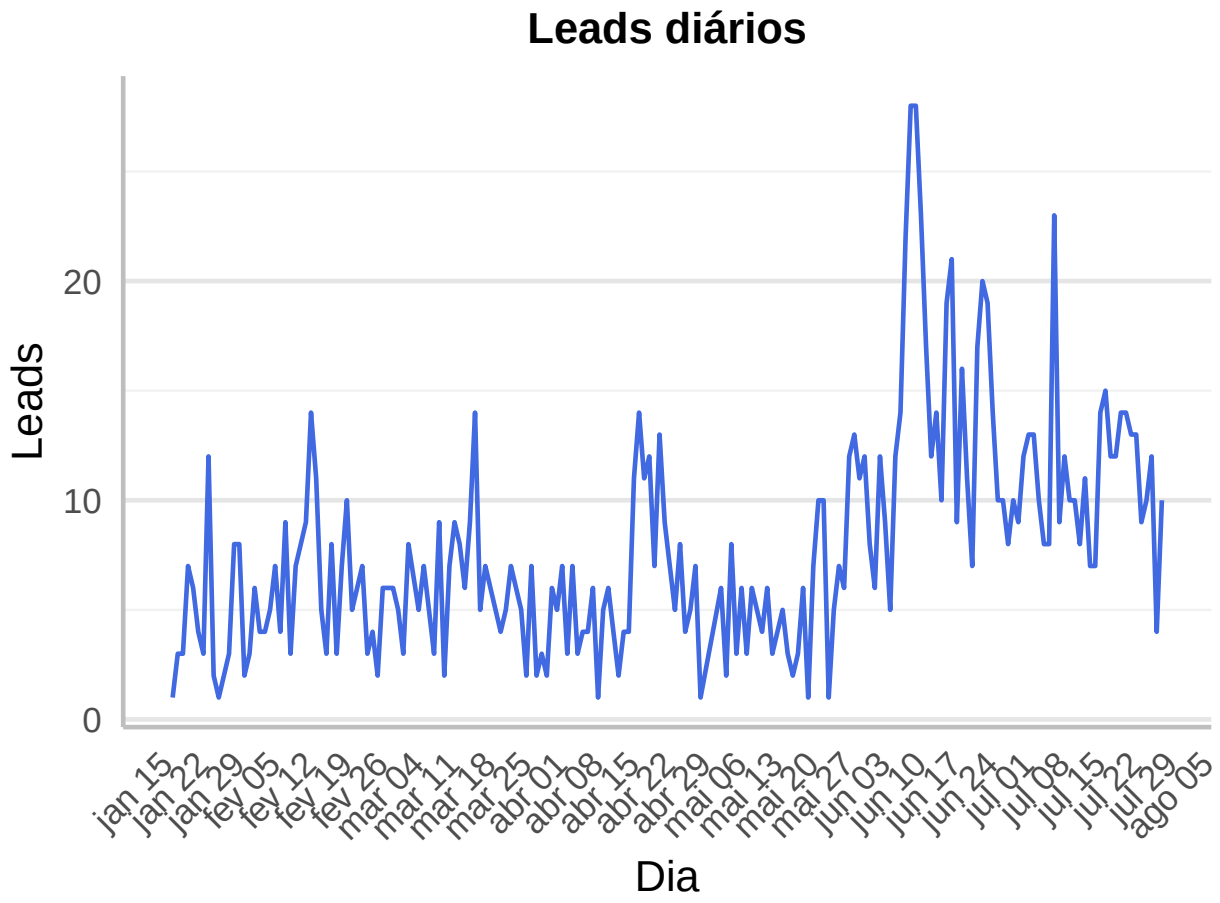
```
df_leads <- read_excel("dataset/leads.xlsx", col_types = c("numeric", "date", "text"))
df_leads$data_criado_em <- as.Date(df_leads$data_criado_em)
kable(head(df_leads))
```

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



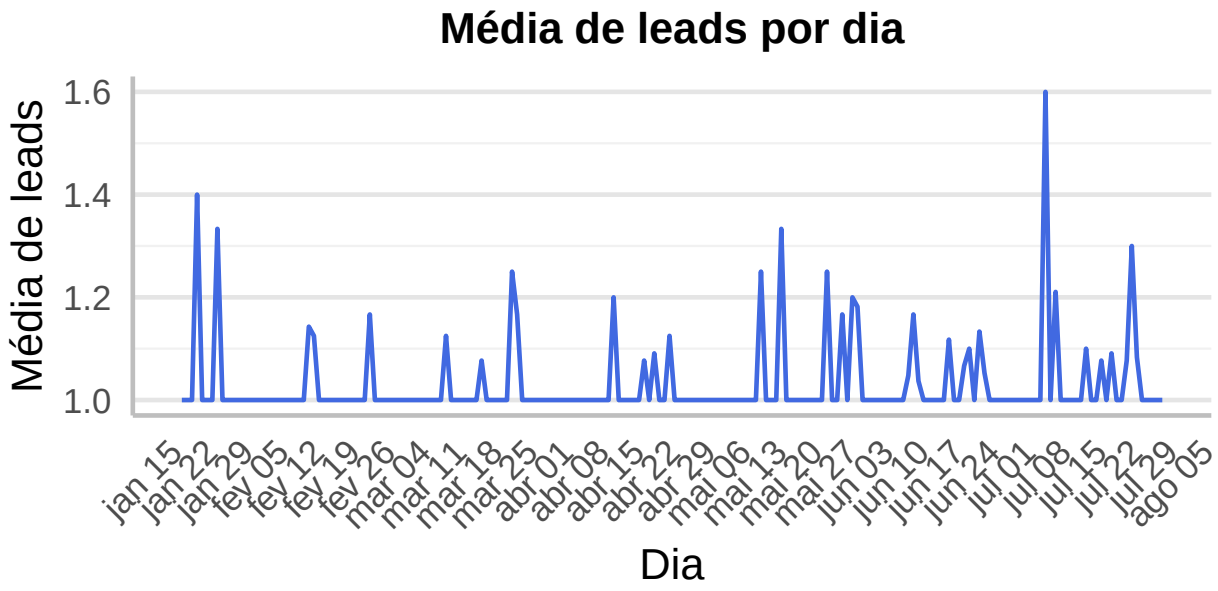
Notam-se picos em intervalos semi-regulares, o que pode indicar sazonalidade ou eventos específicos. Além disso, em Julho, houve uma alta variabilidade nos leads diários.

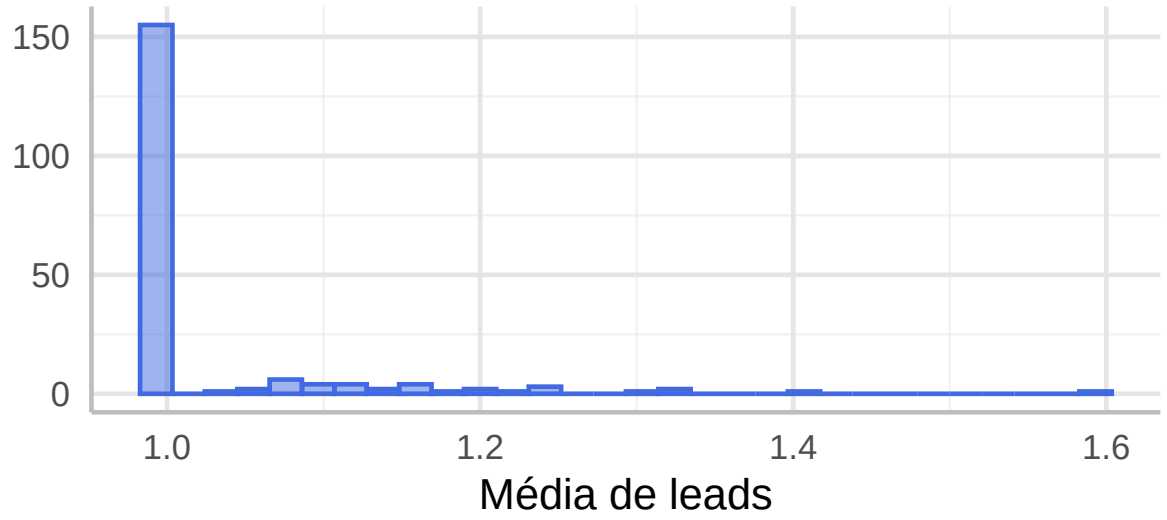
Leads por anúncio Vamos analisar a média diária de leads por anúncio.

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
)
```





Será que é só isso mesmo?

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.

Estimativa da média

Para corrigir o problema anterior, 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.

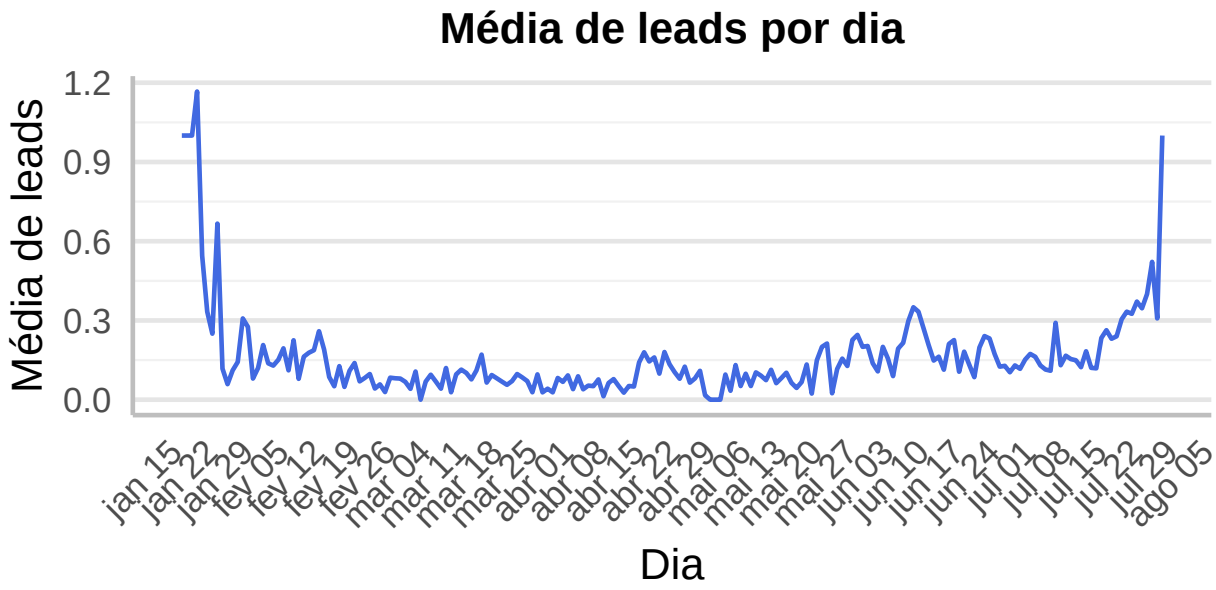
```
df_leads_complete <- df_leads %>%
  group_by(id_anuncio, data_criado_em) %>%
  summarise(leads = n(), .groups = 'drop') %>%
  group_by(id_anuncio) %>%
  # Creates a list of dataframes by id
  tidyr::nest() %>%
  mutate(
    # Creates a sequence of dates by id
    date_seq = map(data, ~seq(min(.$data_criado_em), max(.$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))
```

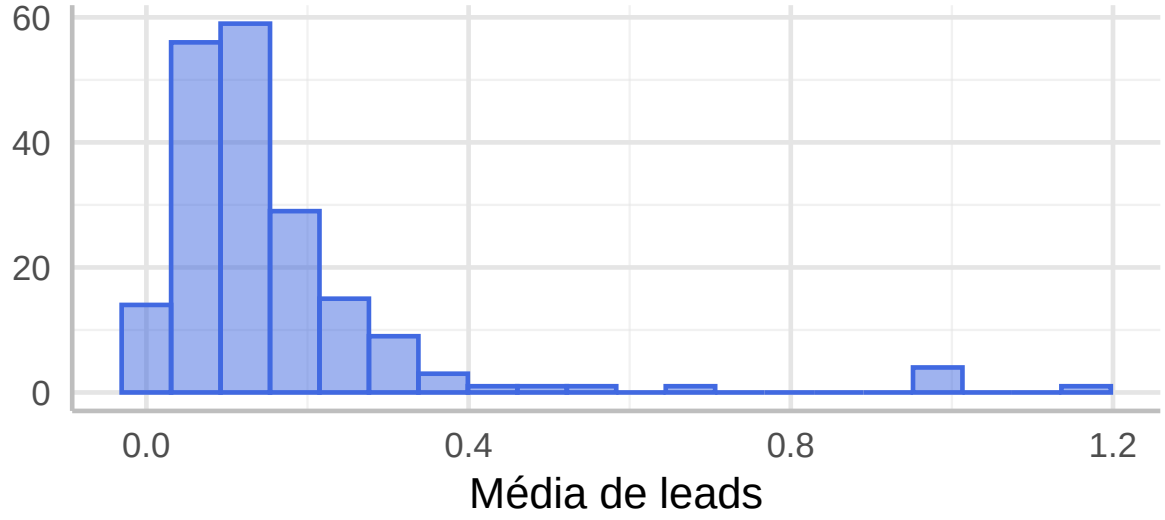
id_anuncio	data_criado_em	leads
00OPP2	2024-03-10	1
00SLR7	2024-06-29	1
00TPRF	2024-06-23	1
02NTL4	2024-04-20	1
02NTL4	2024-04-21	0
02NTL4	2024-04-22	0

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
)
```





Mas não está totalmente correto...

Lembrando que esta é uma aproximação e não corresponde totalmente ao que de fato aconteceu, para computar a verdadeira média, precisariamos 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.

Table 3: Registros corrigidos

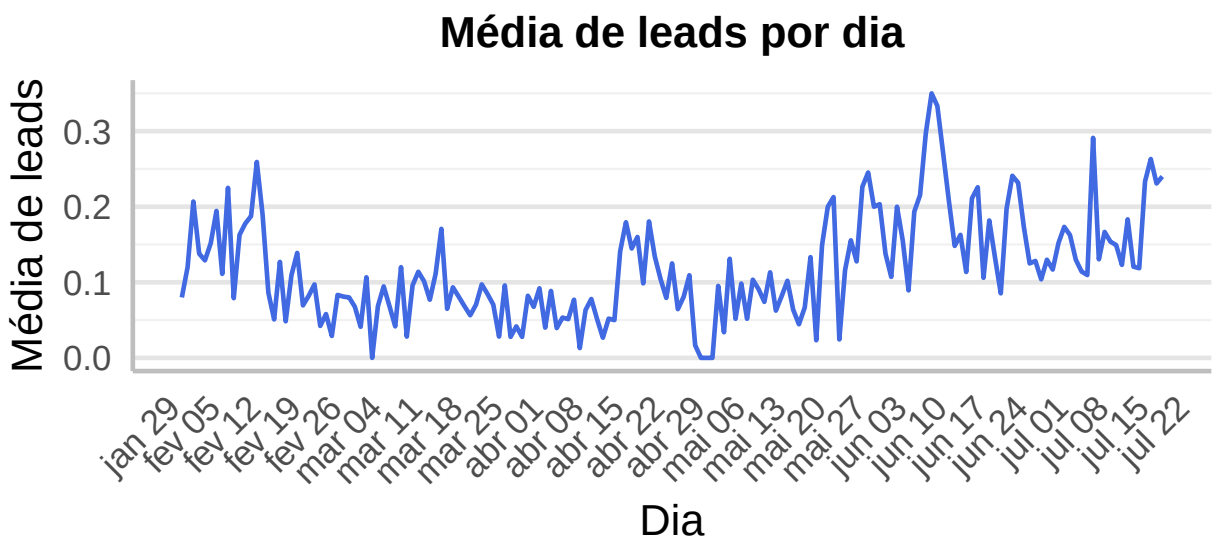
Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 50	Pctl. 75	Max
leads	11253	0.12	0.34	0	0	0	0	4

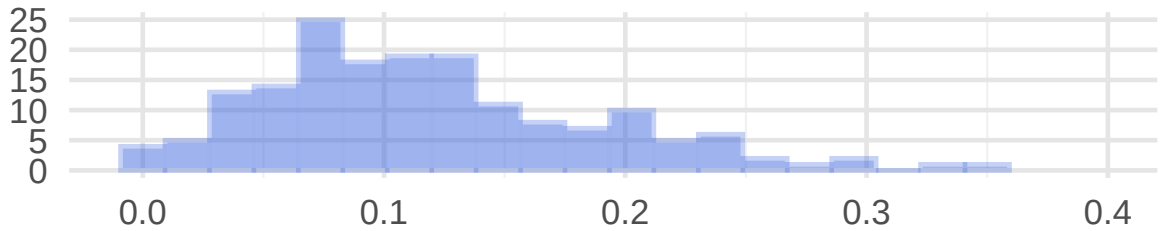
Portanto vamos analisar apenas entre 01/02/2024 e 20/07/2024.

```
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
    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)
)
```






```r
leads_daily_mean <- mean(df_leads_complete_filtered$leads)
leads_daily_sd <- sd(df_leads_complete_filtered$leads)

leads_mean_of_means <- mean(df_leads_daily_filtered$mean)
leads_sd_of_means <- sd(df_leads_daily_filtered$mean)
mean_active_listings <- ceiling(mean(df_leads_daily_filtered$active_listings))

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

Resultados

Assim, apesar dos pesares, temos uma média de `~0.118` leads por dia, com um desvio padrão de `~0.337`. Além disso, a média das médias diárias é de `~0.118` com um desvio padrão de `~0.069`.

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

Tamanho da amostra

Calculamos que o tamanho da amostra, a partir da equação

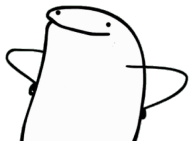
$$n' \geq \left(Z_{\alpha/2}\frac{\sigma}{\mathrm{e}}\right)^2$$
$$n = n' \cdot \frac{N-n}{N-1}$$

onde, $Z_{\alpha/2}$ é o valor crítico da distribuição normal, e é a margem de erro, σ 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 (AASs).

```
confidence <- 0.8
z_quartil <- abs(qnorm((1 - confidence) / 2))
max_error <- 0.05
sigma <- leads_sd_of_means
size_population <- mean_active_listings

computed_sample_size_aasc <- (z_quartil * sigma / max_error)^2
computed_sample_size <- ceiling(
  computed_sample_size_aasc *
    (size_population - computed_sample_size_aasc) / (size_population - 1)
)
```

Lembrando que queremos estimar a média de leads diários, portanto, vamos usar a média das médias diárias.

Ta-dá! Para obter uma margem de erro de `0.05` com `80%` de confiança, utilizando AASs, precisamos de uma amostra de `4` anúncios.



Análise do erro

Vamos analisar o erro da média de leads diários, a partir do tamanho da amostra calculado.

```
sample_repetitions <- 20

df_leads_daily_error <- df_leads_complete_filtered %>%
  group_by(data_criado_em) %>%
  summarise(lead_mean = mean(leads),
            leads = list(leads),
            active_listings = n_distinct(id_anuncio)) %>%
  rowwise() %>%
  mutate(
```

Table 5: Erro da média de leads diários

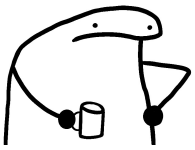
Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 50	Pctl. 75	Max
Média real	171	0.12	0.069	0	0.068	0.11	0.16	0.35
Média das médias amostrais	171	0.12	0.083	0	0.062	0.1	0.16	0.51
Erro médio	171	0.13	0.049	0	0.099	0.13	0.15	0.37

```
    samples = list(replicate(sample_repetitions, sample(leads, computed_sample_size, replace = F), simpl
    samples_mean = list(map_dbl(samples, mean)),
    samples_error = list(map_dbl(samples_mean, ~ abs(.x - lead_mean))),
    min_mean = min(samples_mean),
    max_mean = max(samples_mean),
    mean_mean = mean(samples_mean),
    min_error = min(samples_error),
    max_error = max(samples_error),
    mean_error = mean(samples_error)
  ) %>%
  select(-samples, -samples_mean, -samples_error, -leads)

sample_mean_means <- mean(df_leads_daily_error$mean_mean)
sample_mean_errors <- mean(df_leads_daily_error$mean_error)

colnames_lookup <- c(
  "Média real" = "lead_mean",
  "Média das médias amostrais" = "mean_mean",
  "Erro médio" = "mean_error"
)
sumtable(df_leads_daily_error %>%
           select(lead_mean, mean_mean, mean_error) %>%
           rename(all_of(colnames_lookup)),
         add.median = T, title = "Erro da média de leads diários")
```

Olhando a tabela acima, vemos que, com uma amostra de `4` anúncios, a média das médias amostrais de leads diários é de `0.12` leads, com um erro médio de `0.127` leads.

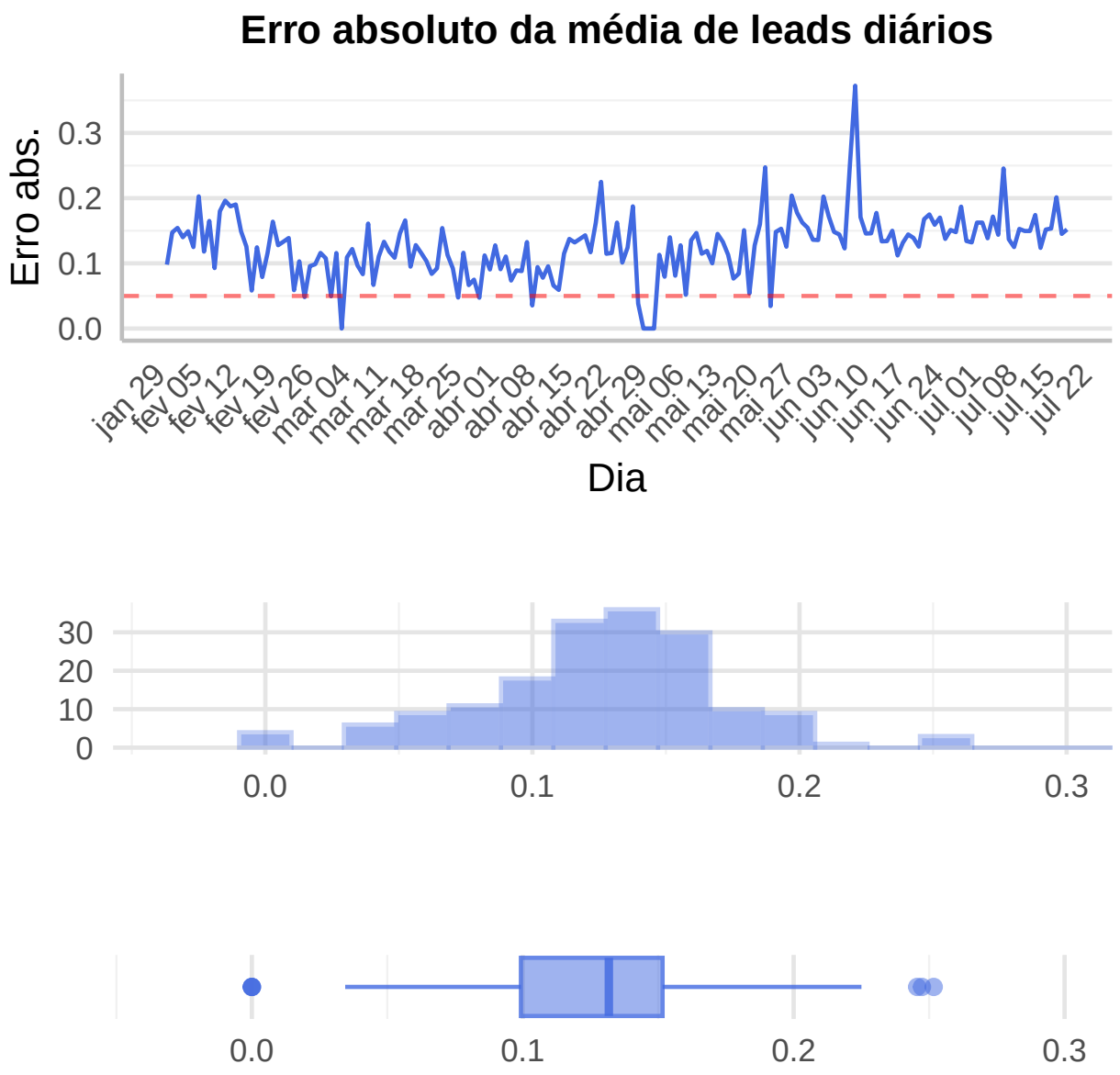
Ou seja, o valor é bem próximo da média real, no entanto, o erro médio é bem maior `0.05`.

Mas pera aí...



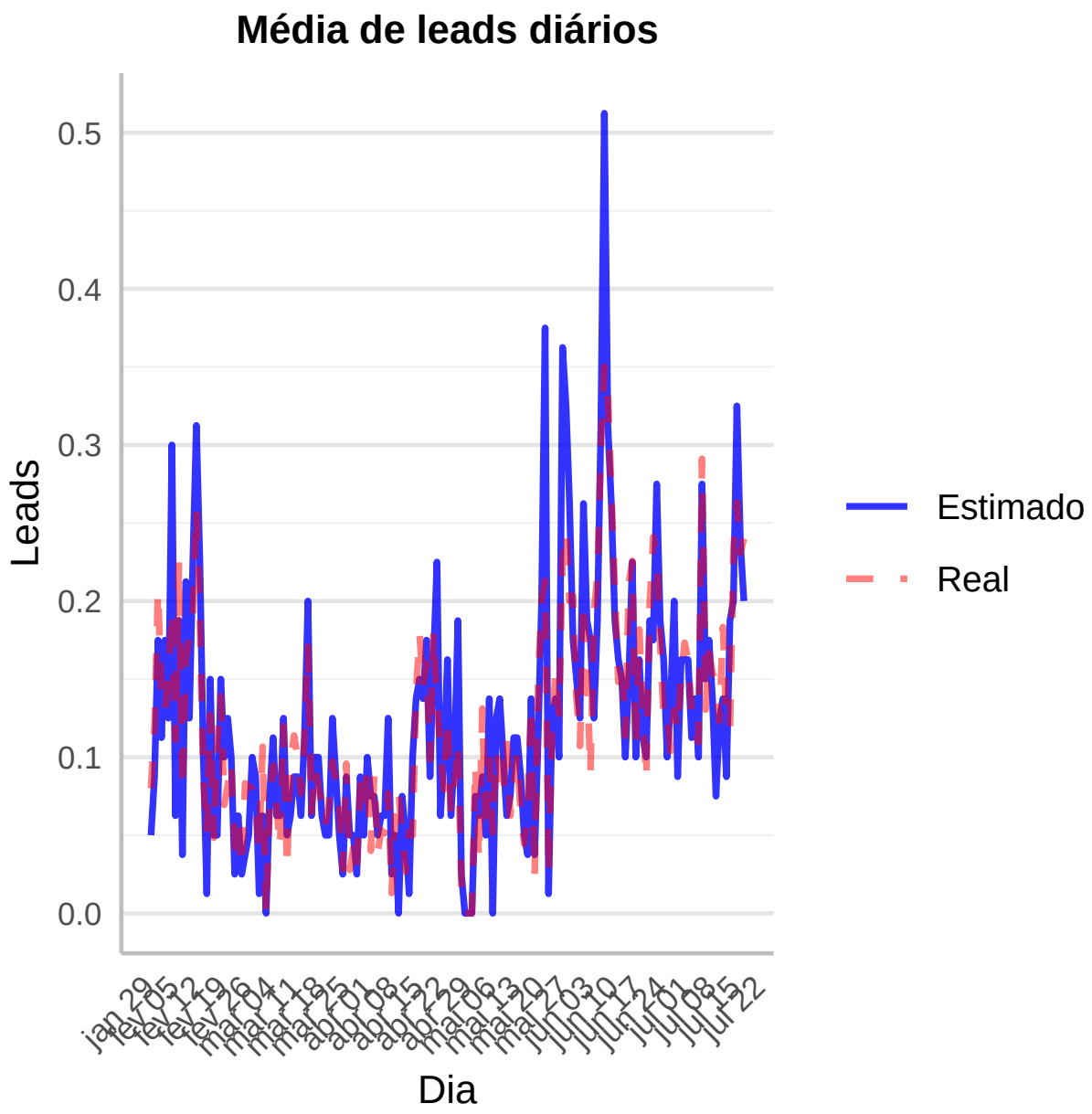
```
grid.arrange(
  df_leads_daily_error %>%
    ggplot(aes(data_criado_em, mean_error)) +
    geom_line(color = "royalblue", linewidth = 0.5) +
    geom_hline(yintercept = max_error, color = "red", linetype = "dashed", alpha = 0.5) +
    labs(title = "Erro absoluto da média de leads diários",
         x = "Dia",
         y = "Erro abs.") +
    apply.theme.ts(),
```

```
  df_leads_daily_error %>%
    ggplot(aes(mean_error)) +
    coord_cartesian(xlim = c(-0.04, 0.3)) +
    geom_histogram(bins = 20, color = adjustcolor("royalblue", alpha.f = 0.3), fill = "royalblue", alpha
    labs(title = "",
         x = "",
         y = "") +
    theme.base + theme.no_legend + theme.no_axis +
    theme(panel.grid.minor.y = element_blank()),
  df_leads_daily_error %>%
    ggplot(aes(mean_error)) +
    coord_cartesian(xlim = c(-0.05, 0.3)) +
    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)
)
```



Apesar de parecer correto quando analizamos o todo, o gráfico nos mostra que estamos errando além do esperado a maior parte das vezes. Isso indica que a amostra não é suficiente para garantir a precisão desejada.

```
df_leads_daily_error %>%
  ggplot(aes(data_criado_em)) +
  geom_line(aes(y = mean_mean, size = "Estimado"), color = "blue", alpha = 0.8, linewidth = 0.8) +
  geom_line(aes(y = lead_mean, size = "Real"), color = "red", linetype = "dashed", linewidth = 0.8, alpl
  labs(title = "Média de leads diários",
       x = "Dia",
       y = "Leads",
       size = "") +
  apply.theme.ts.legend()
```



Mas então, o que pode ter dado errado?

Causas do erro

- **Variabilidade dos dados**: para calcular o tamanho da amostra utilizamos a média de todas as médias diárias, desta forma, perdemos a informação de que há dias com alta variabilidade.
- **Independência**: outro fator que influencia nesse erro é a falta de independência entre os dados. Isto é, os dados utilizados formam uma série temporal, o que faz com que qualquer estatística utilizada seja dependente do tempo. Isso torna incorreta a técnica de estimação do tamanho da amostra, pois tem como premissa a independência dos dados.

Sugestão de tamanho da amostra

Para trabalhos futuros sugerimos que **não** utilize as técnicas de estimação de tamanho de amostra apresentadas neste trabalho. Em vez disso, sugerimos que sejam utilizadas técnicas de amostragem próprias para séries temporais.