lab3-ckp1

Fernando Lordao

16 de maio de 2018

```
running_mode = "finalWeb"
#running_mode = "reducedLocal"
#running_mode = "finalLocal"

if(running_mode == "finalWeb") {
    path_to_read = "https://github.com/wikimedia-research/Discovery-Hiring-Analyst-2016/raw/master/event
    lines_to_read = -1
} else if(running_mode == "finalLocal") {
    path_to_read = "data/events_log.csv"
    lines_to_read = -1
} else if(running_mode == "reducedLocal") {
    path_to_read = "data/events_log.csv"
    lines_to_read = 5000
} else {
    message("Running mode not specified.")
    break() #encontrar a função certa para finalizar o script.
}

set.seed(20180520)
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.2     v dplyr   0.7.4
## v tidyr   0.8.0     v stringr 1.3.0
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts -------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(chron)
```

```
##
## Attaching package: 'chron'
```

```
## The following objects are masked from 'package:lubridate':
##
##     days, hours, minutes, seconds, years
```

```
library(boot)
theme_set(theme_bw())
```

Descrição geral

Neste documento vamos explorar os dados disponibilizados pela Wikimedia e tentar, no percurso dessa análise exploratória, responder às questões levantadas fazendo inferências sobre a população através de bootstraps.

Carga do log de eventos

```
events = read_csv(path_to_read) %>%
    head(lines_to_read)
```

```
## Parsed with column specification:
## cols(
##   uuid = col_character(),
##   timestamp = col_double(),
##   session_id = col_character(),
##   group = col_character(),
##   action = col_character(),
##   checkin = col_integer(),
##   page_id = col_character(),
##   n_results = col_integer(),
##   result_position = col_integer()
## )
```

```
#ATTENTION! Não entendi o uso do slice
#events = events %>% slice(1:5e4) # Útil para testar código em dados pequenos. Comente na hora de proce

events = events %>%
    mutate(
        date = round_date(ymd_hms(timestamp), unit = "day"),
        date_week = paste(date, weekdays(date, abbreviate = TRUE))
    )
```

PERGUNTA 1:

Qual a taxa geral de cliques durante o dia? Como ela varia entre os grupos?

Conforme descrição do problema pela Wikimedia, a taxa de cliques é definida da seguinte maneira:

clickthrough rate: the proportion of search sessions where the user clicked on one of the results displayed

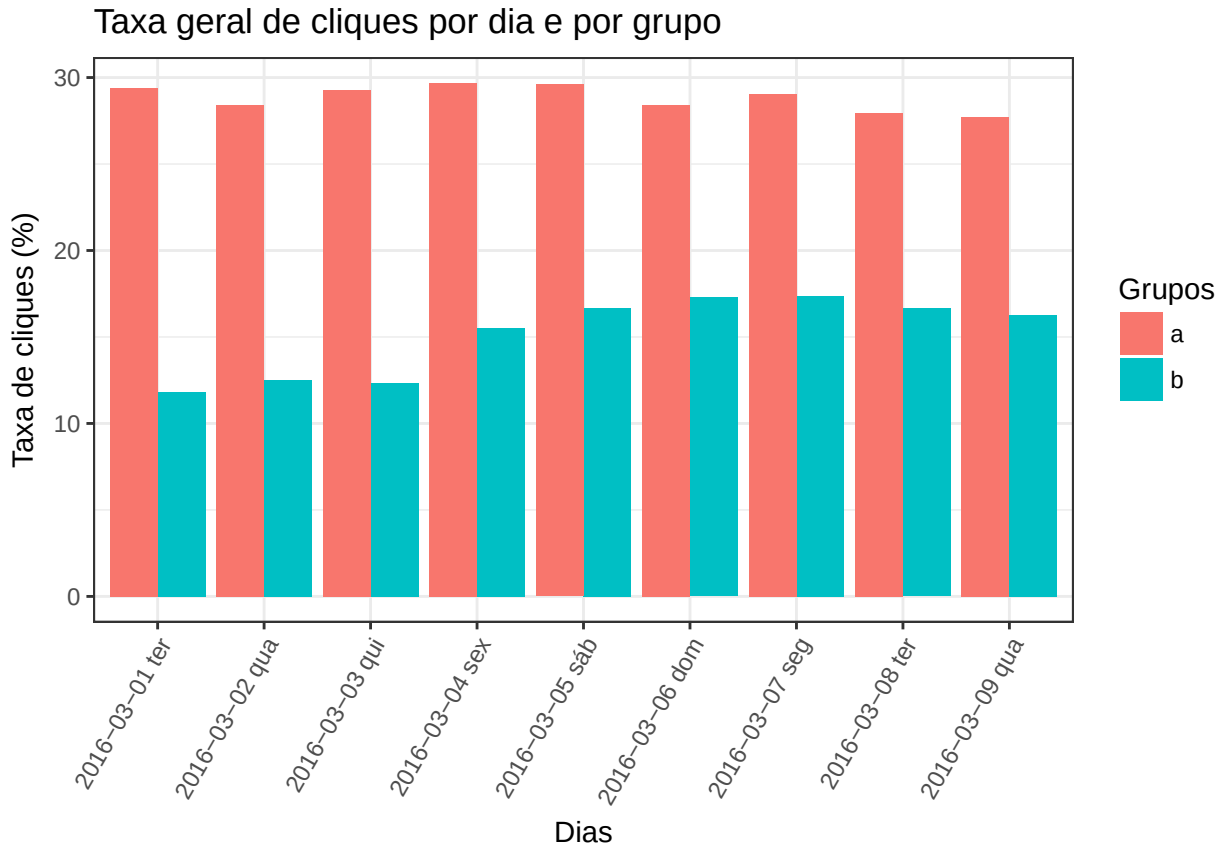
Para isso é preciso criar novas variáveis que permitam identificar as buscas e também as visitas dentre de cada busca. Faremos isso usando o código a seguir.

```
events = events %>%
    group_by(session_id) %>%
    arrange(timestamp) %>%
    mutate(
        search_index = cumsum(action == "searchResultPage") # Sequenciador de buscas realizadas dentro
    )
```

```r
events = events %>%
    group_by(session_id, search_index) %>%
    arrange(timestamp) %>%
    mutate(
        visit_index = cumsum(action == "visitPage") # Sequenciador de visitas realizadas dentro de cada
    )
```

Segue código para produzir os dados diários, por grupo, da *clickthrough rate* considerando a noção definida pela Wikimedia.

```r
events_q1 = events %>%
  group_by(group, date_week, session_id, search_index) %>%
  summarize(visit_count = max(visit_index)) %>%
  group_by(group, date_week) %>%
  summarize(
    nonzero_click = sum(visit_count>0),
    zero_click = sum(visit_count==0),
    clickthroughrate = nonzero_click / (nonzero_click + zero_click)
  )


events_q1 %>%
    ggplot(aes(x = date_week, y = clickthroughrate*100, fill = group)) +
    geom_col(position = "dodge") +
    labs(x = "Dias", y = "Taxa de cliques (%)", fill = "Grupos", title = "Taxa geral de cliques por dia
    theme(axis.text.x = element_text(angle=60, hjust=1))
```



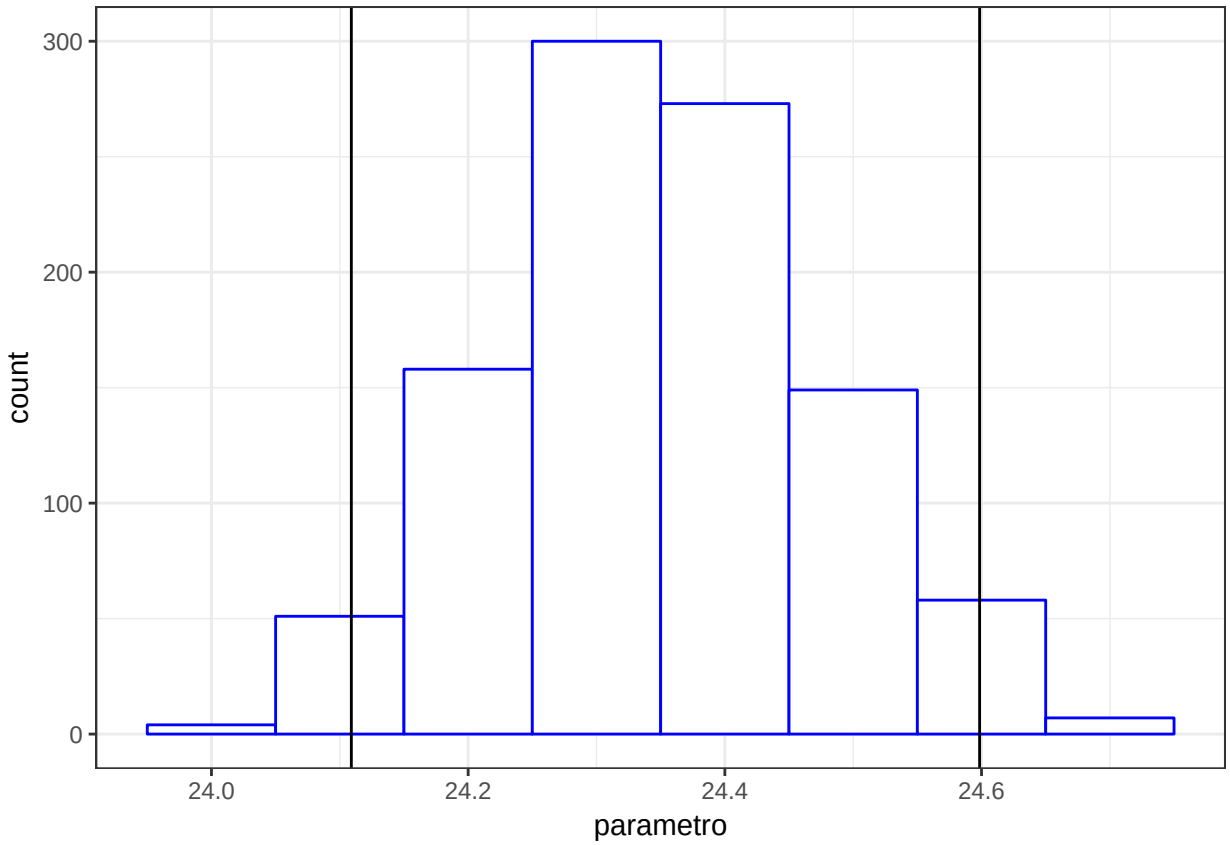
Analisando a média de taxa de cliques diária geral (sem distinção de grupos)

```
funcao_bootstrap_q1_global <- function(data, indexes){
  daily_rate = data %>%
    slice(indexes) %>%
    group_by(date_week) %>%
    summarize(
      clickthroughrate = (sum(click_count>0) / sum(click_count>=0)) * 100
    )
  return(
    mean(daily_rate$clickthroughrate)
  )
}

events_q1b = events %>%
    group_by(date_week, session_id, search_index) %>%
    mutate(click_count = max(visit_index)) %>%
    subset(action == "searchResultPage", select = c("date_week", "click_count"))


bootstraps_q1 <- boot(data = events_q1b,
                  statistic = funcao_bootstrap_q1_global, # <- referência para a função
                  R = 1000) # número de bootstraps

glimpse(bootstraps_q1$t)
```

```
##  num [1:1000, 1] 24.4 24.6 24.4 24.2 24.3 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q1$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (24.10, 24.59 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a taxa de cliques da população está entre 29,10% e 29,77%.

Analisando a média de taxa de cliques diária geral (do grupo A)

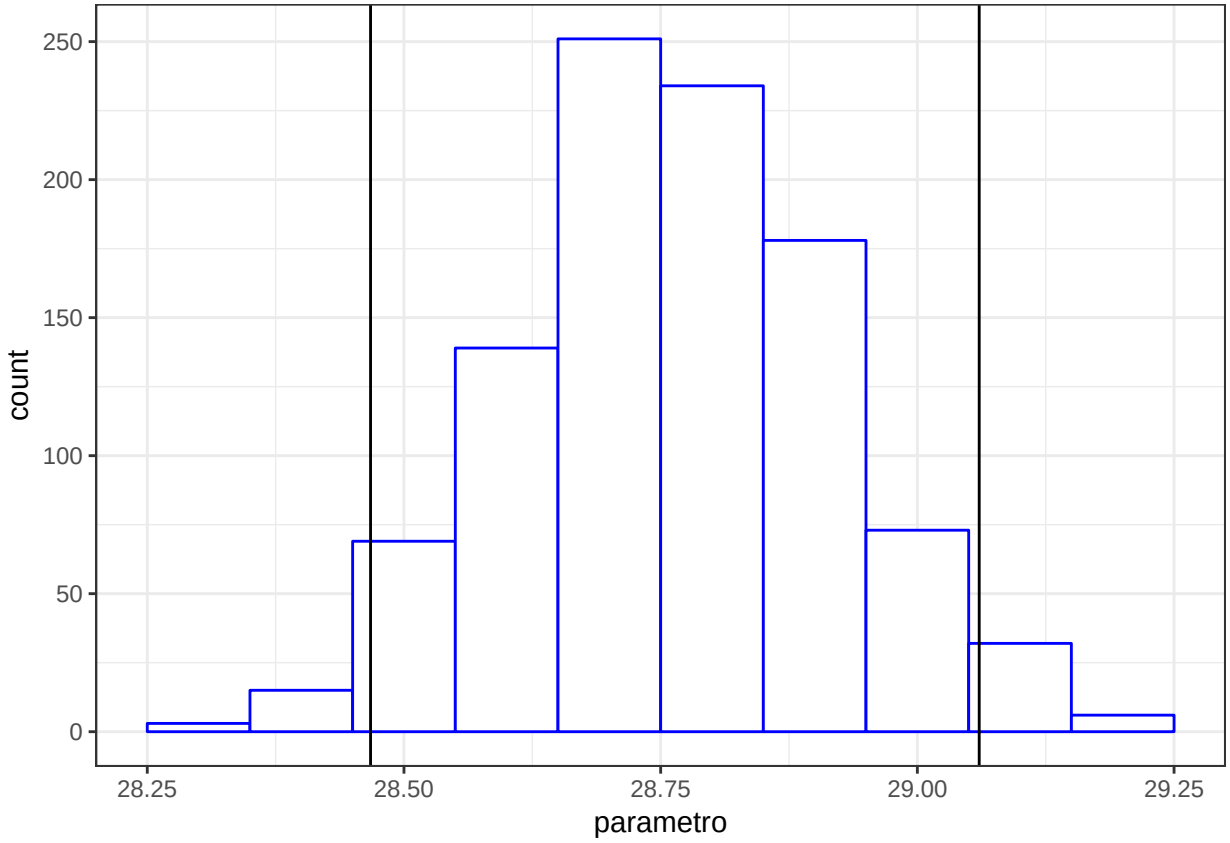
```
events_q1ba = events %>%
    filter(group == "a") %>%
    group_by(date_week, session_id, search_index) %>%
    mutate(click_count = max(visit_index)) %>%
    subset(action == "searchResultPage", select = c("date_week", "click_count"))
```

```
bootstraps_q1 <- boot(data = events_q1ba,
                      statistic = funcao_bootstrap_q1_global, # <- referência para a função
                      R = 1000) # número de bootstraps

glimpse(bootstraps_q1$t)
```

```
##  num [1:1000, 1] 28.3 28.6 28.7 28.9 28.7 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q1$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
```

```
## boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (28.46, 29.06 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a taxa de cliques no grupo A da população está entre 28,46% e 29,06%.

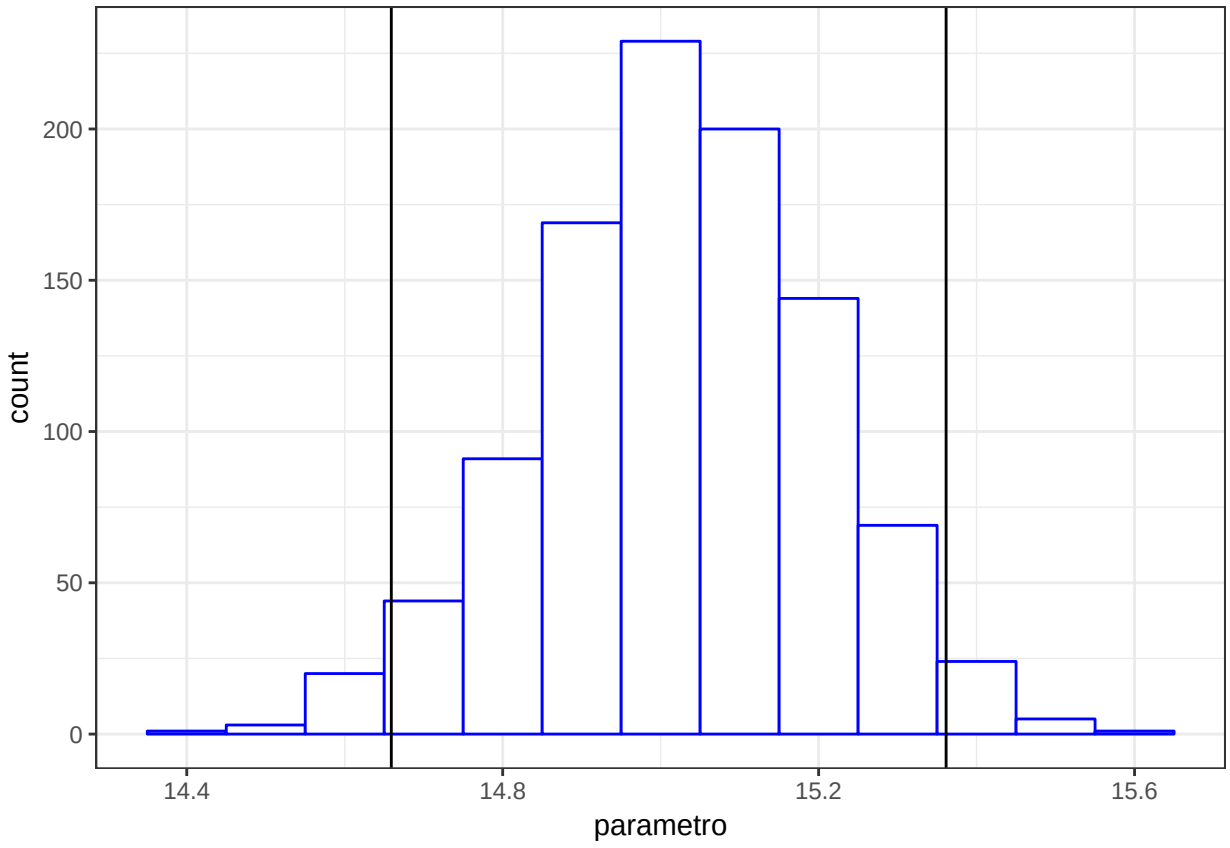
Analisando a média de taxa de cliques diária geral (do grupo B)

```
events_q1bb = events %>%
    filter(group == "b") %>%
    group_by(date_week, session_id, search_index) %>%
    mutate(click_count = max(visit_index)) %>%
    subset(action == "searchResultPage", select = c("date_week", "click_count"))


bootstraps_q1 <- boot(data = events_q1bb,
                  statistic = funcao_bootstrap_q1_global, # <- referência para a função
                  R = 1000) # número de bootstraps

glimpse(bootstraps_q1$t)
```

```
##  num [1:1000, 1] 15.4 14.9 15.2 14.8 15.3 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q1$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (14.71, 15.41 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a taxa de cliques no grupo A da população está entre 14,72% e 15,36%.

Analisando a diferença da média de taxa de cliques diária entre o grupo A e o grupo B

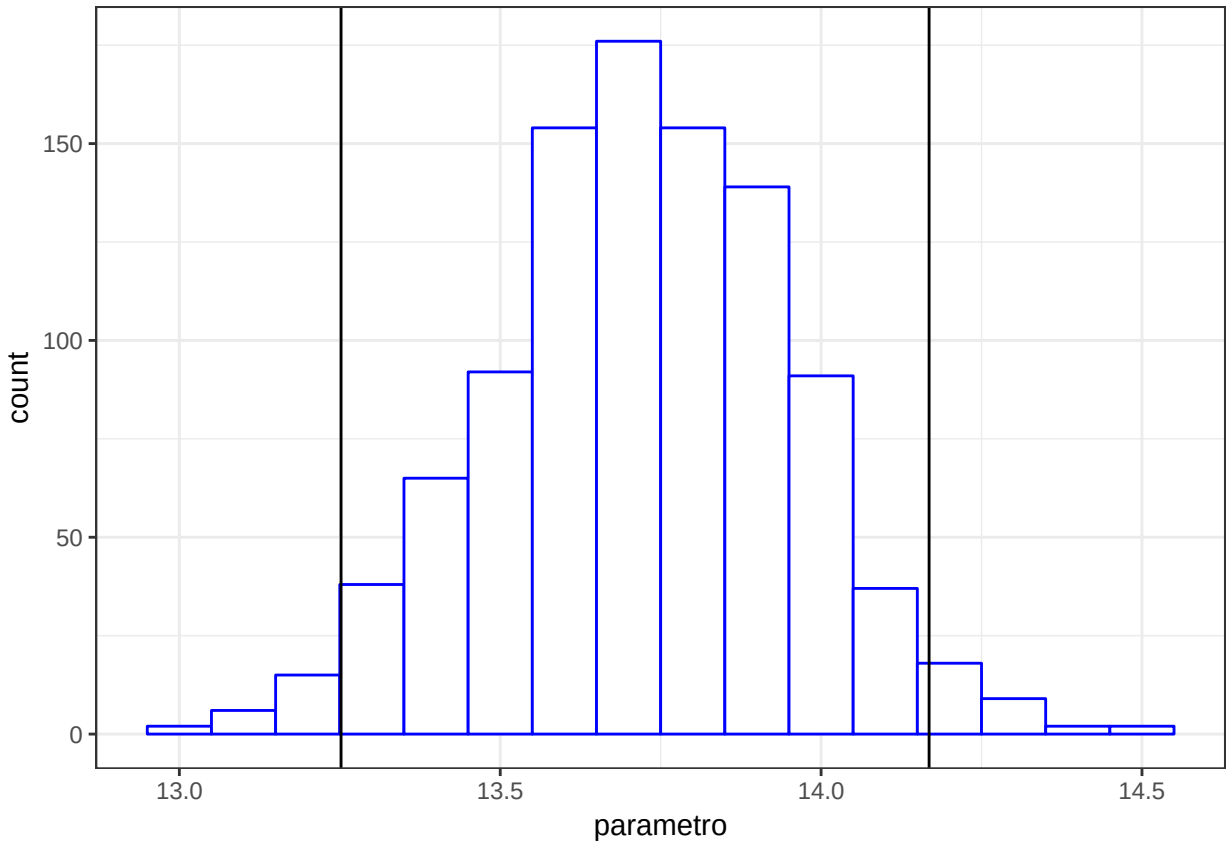
```
events_q1bdif = events %>%
    group_by(group, date_week, session_id, search_index) %>%
    mutate(click_count = max(visit_index)) %>%
    subset(action == "searchResultPage", select = c("group", "date_week", "click_count"))
```

```r
funcao_bootstrap_q1_dif <- function(data, indexes){
  daily_rate = data %>%
    slice(indexes) %>%
    group_by(group, date_week) %>%
    summarize(
      clickthroughrate = (sum(click_count>0) / sum(click_count>=0)) * 100
    )
  mean_rate_a = daily_rate %>%
    filter(group == "a") %>%
    summarize(
      meanrate = mean(clickthroughrate)
    ) %>%
    pull (meanrate)

  mean_rate_b = daily_rate %>%
    filter(group == "b") %>%
    summarize(
      meanrate = mean(clickthroughrate)
    ) %>%
    pull (meanrate)

  return(
    mean(mean_rate_a - mean_rate_b)
  )
}

bootstraps_q1 <- boot(data = events_q1bdif,
                      statistic = funcao_bootstrap_q1_dif, # <- referência para a função
                      R = 1000) # número de bootstraps

glimpse(bootstraps_q1$t)
```

```
##  num [1:1000, 1] 13.1 13.8 13.6 13.6 14 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```r
tibble(parametro = as.double(bootstraps_q1$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q1$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q1, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (13.28, 14.20 )
## Calculations and Intervals on Original Scale
```

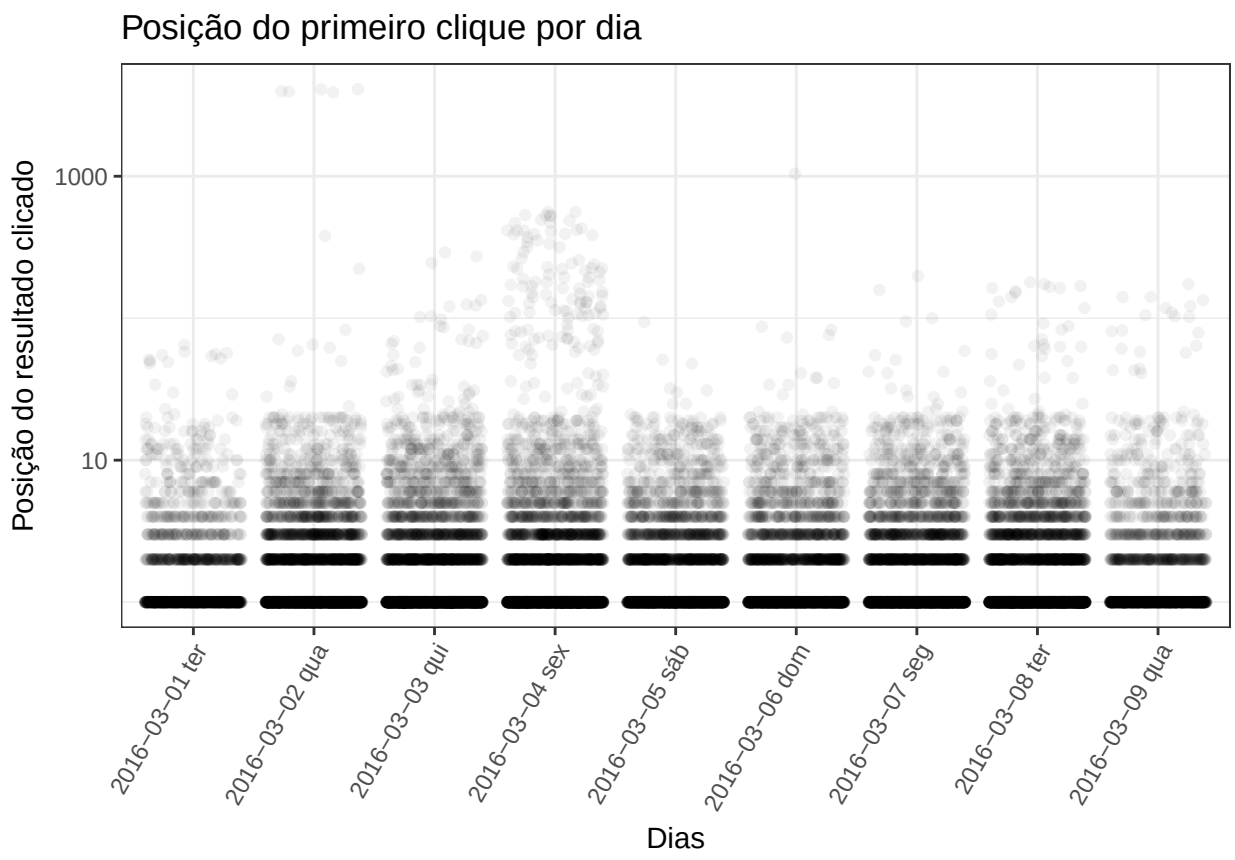
Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a média da taxa de cliques diária no grupo A está acima da taxa média do grupo B num valor entre 13,28 e 14,18 pontos percentuais.

PERGUNTA 2:

Quais resultados os usuários tendem a clicar primeiro? Como isso muda dia após dia?

Análise original.

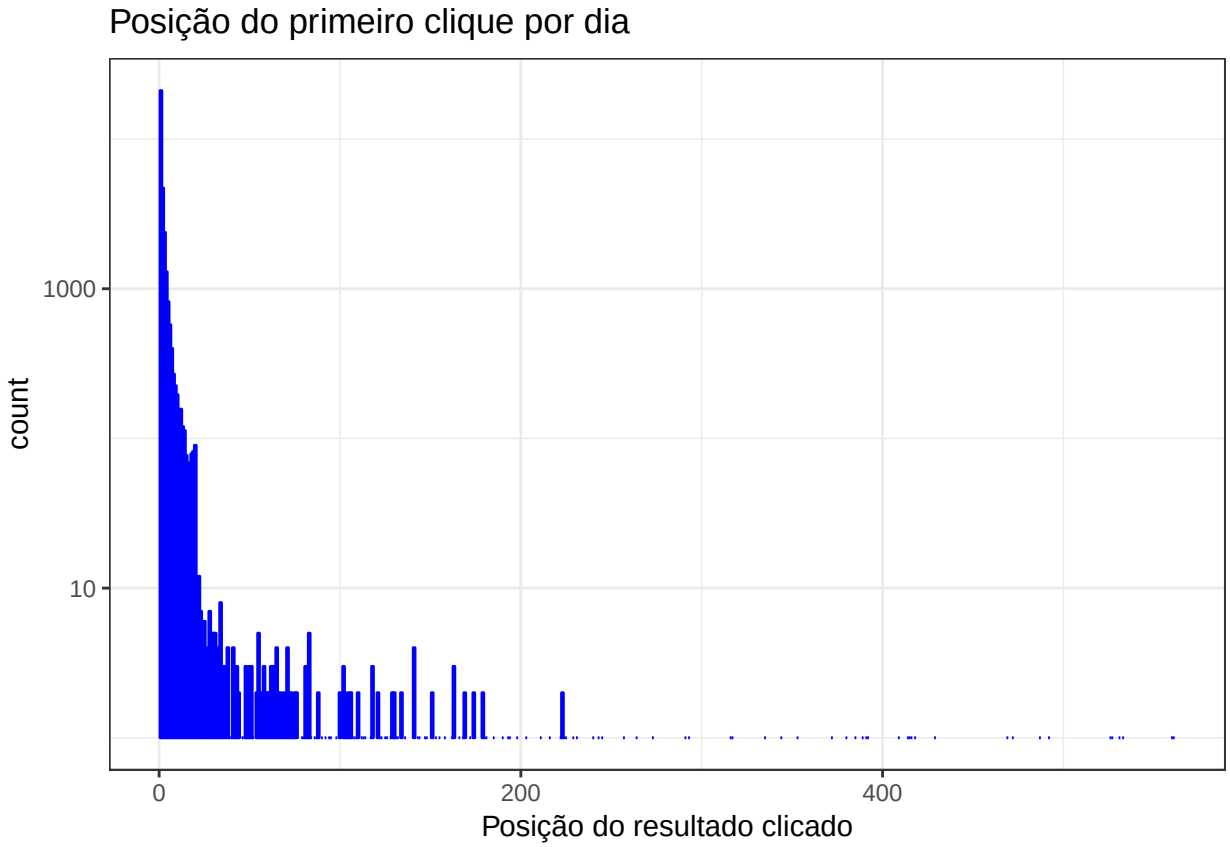
```
events_q2 = events %>%
    filter(visit_index == 1 & action == "visitPage" & !is.na(result_position))

events_q2 %>%
    ggplot(aes(x = date_week, y = result_position)) +
    geom_jitter(alpha = 0.05) +
    labs(x = "Dias", y = "Posição do resultado clicado", title = "Posição do primeiro clique por dia") +
    scale_y_log10() +
    theme(axis.text.x = element_text(angle=60, hjust=1))
```




```
events_q2 %>%
    filter(result_position < 1000) %>%
    ggplot(aes(x = result_position)) +
    geom_histogram(binwidth = 1, fill = "white", color = "blue", show.legend = TRUE) +
    labs(x = "Posição do resultado clicado", title = "Posição do primeiro clique por dia") +
    scale_y_log10()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 378 rows containing missing values (geom_bar).
```



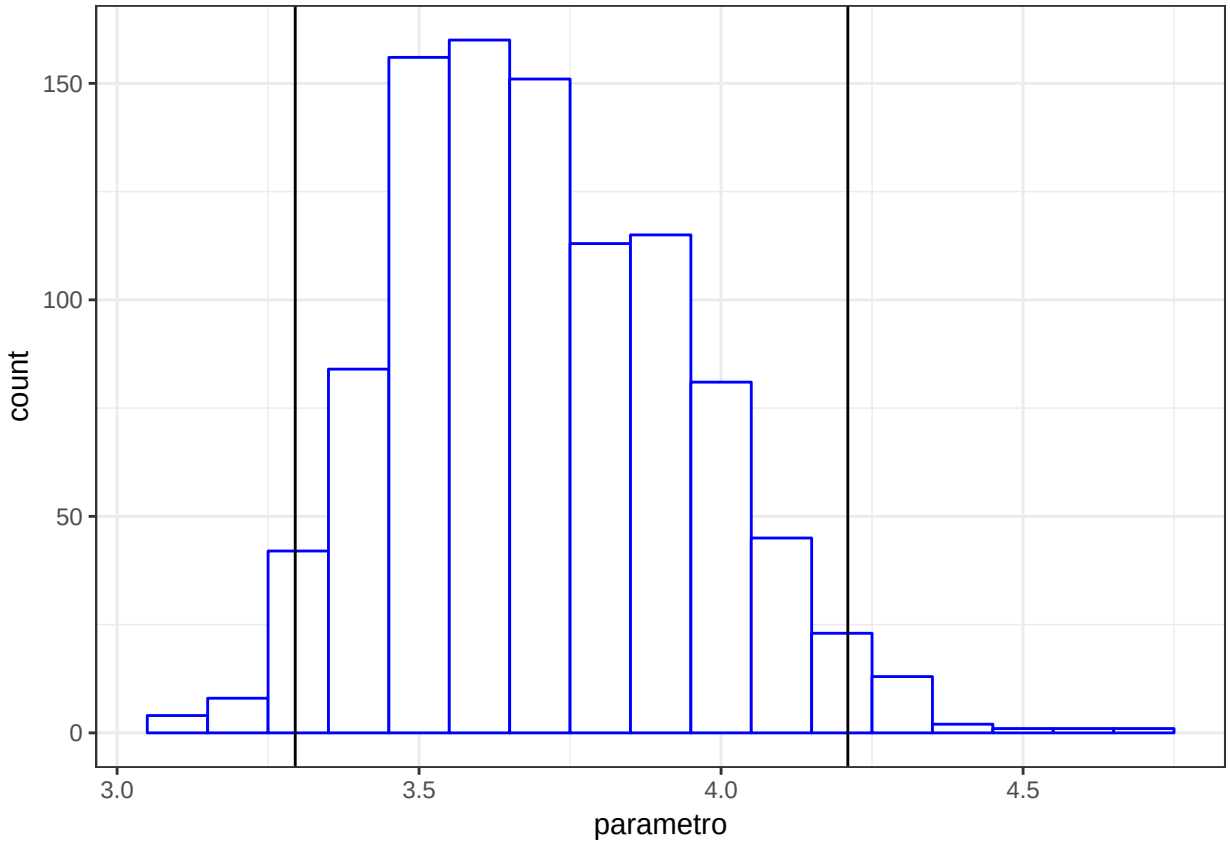
Analisando a média de taxa de cliques diária geral

```
funcao_bootstrap_q2_global <- function(data, indexes){
  daily_data = data %>%
    slice(indexes) %>%
    group_by(date_week) %>%
    summarize(
      mean_position = mean(result_position)
    )
  return(
    mean(daily_data$mean_position)
  )
}

events_q2b = events_q2 %>%
    subset(select = c("date_week", "result_position"))


bootstraps_q2b <- boot(data = events_q2b,
                   statistic = funcao_bootstrap_q2_global, # <- referência para a função
                   R = 1000) # número de bootstraps

glimpse(bootstraps_q2b$t)
```

```
##  num [1:1000, 1] 3.51 3.56 3.83 3.32 3.59 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q2b$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q2b$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q2b$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q2b, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q2b, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   ( 3.192,  4.113 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que os usários tendem a clicar nos resulados que estão entre as posições 3,173 e 4,124.

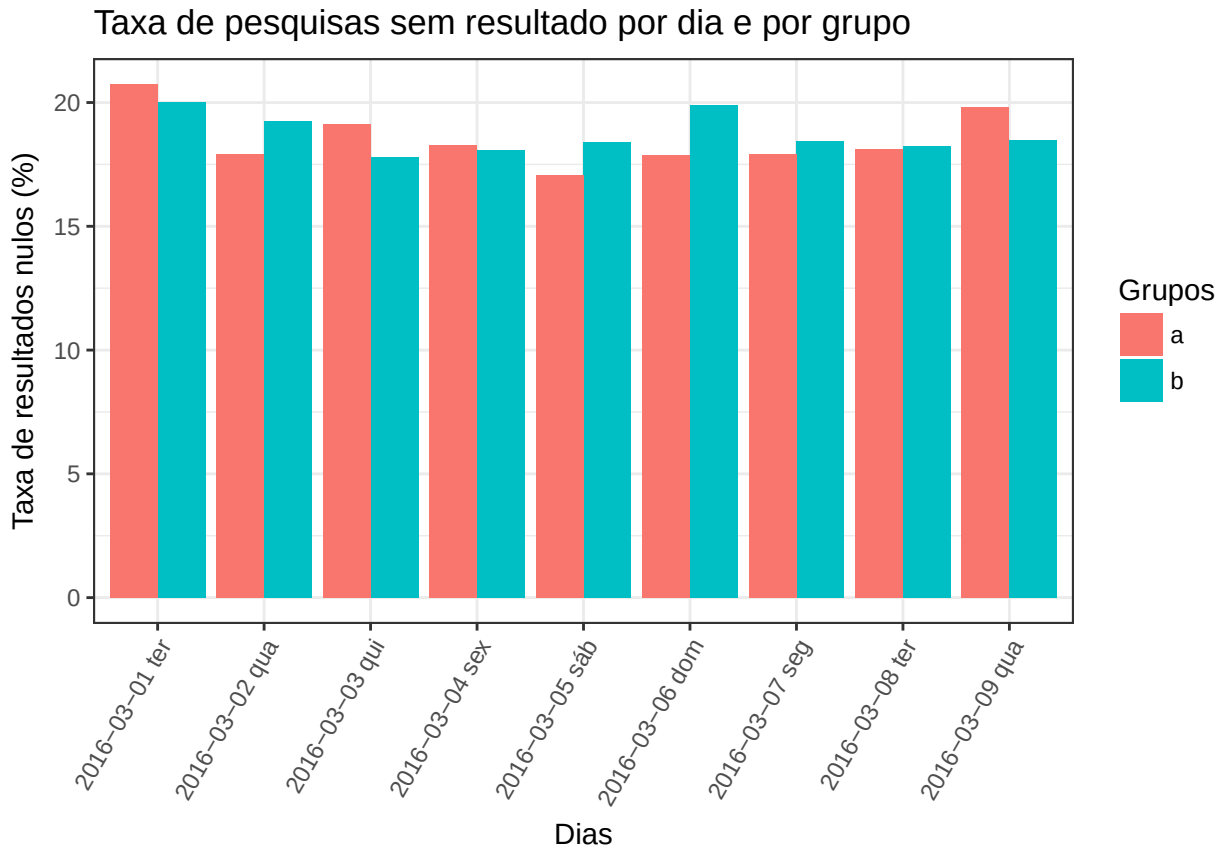
PERGUNTA 3:

Qual a nossa taxa geral de pesquisas sem resultados diariamente? Como isso varia entre os grupos?

Para responder esta questão precisamos contar a quantidade eventos do tipo "searchResultPage" cuja coluna *n_result* é zero e dividir pela quantidade total de eventos desse mesmo tipo.

O gráfico que segue é bem semelhante ao gráfico plotado na questão 1, só muda a informação que estamos medindo, que aqui é a taxa de resultados nulos, por assim dizer.

```
events_q3 = events %>%
    group_by(group, date_week) %>%
    summarise(
        zero_count = sum(action == "searchResultPage" & n_results == 0),
        tot_count = sum(action == "searchResultPage"),
        zerorate = zero_count/tot_count*100
    )

events_q3 %>%
    ggplot(aes(x = date_week, y = zerorate, fill = group)) +
    geom_col(position = "dodge") +
    labs(x = "Dias", y = "Taxa de resultados nulos (%)", fill = "Grupos", title = "Taxa de pesquisas ser
    theme(axis.text.x = element_text(angle=60, hjust=1))
```



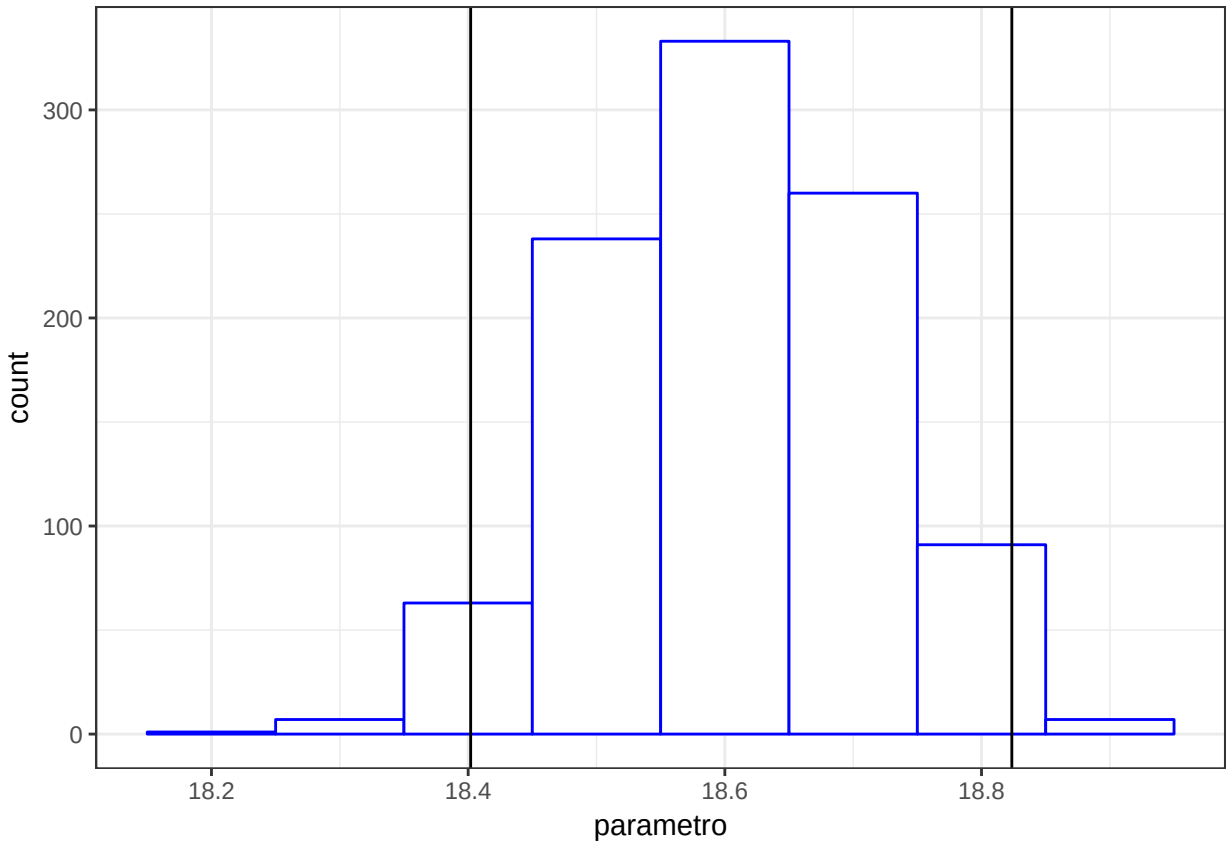
Analisando a média de taxa de pesquisas sem resultado geral (sem distinção de grupos)

```
funcao_bootstrap_q3_global <- function(data, indexes){
  daily_rate = data %>%
    slice(indexes) %>%
    group_by(date_week) %>%
    summarize(
      zero_rate = (sum(n_results==0) / sum(n_results>=0)) * 100
    )
  return(
    mean(daily_rate$zero_rate)
  )
}

events_q3b = events %>%
    group_by(date_week, session_id, search_index) %>%
    subset(action == "searchResultPage", select = c("date_week", "n_results"))


bootstraps_q3 <- boot(data = events_q3b,
                  statistic = funcao_bootstrap_q3_global, # <- referência para a função
                  R = 1000) # número de bootstraps

glimpse(bootstraps_q3$t)
```

```
##  num [1:1000, 1] 18.7 18.6 18.5 18.5 18.6 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q3$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (18.40, 18.83 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a taxa de pesquisas sem resultado está entre 18,40% e 18,83%.

Analisando a média de taxa de pesquisas sem resultados diária geral (do grupo A)

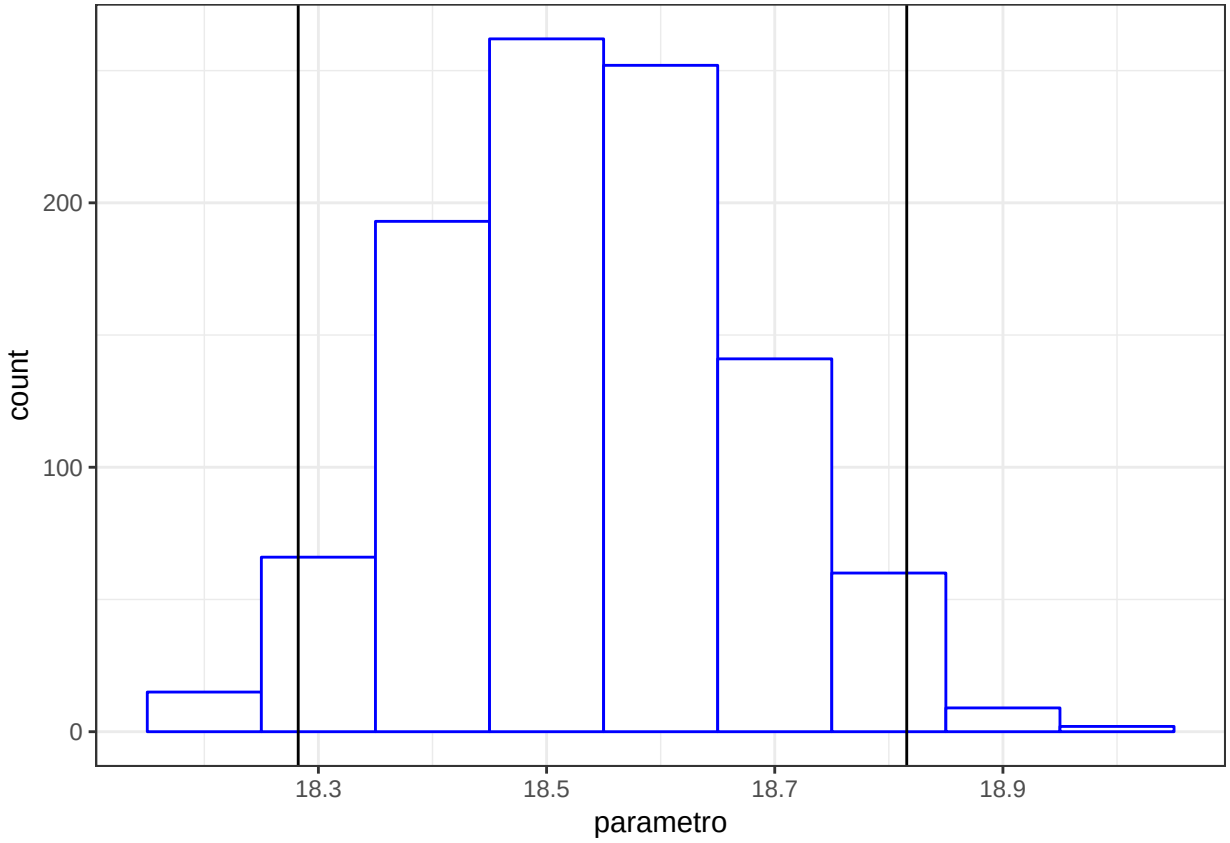
```
events_q3ba = events %>%
    filter(group == "a") %>%
    group_by(date_week, session_id, search_index) %>%
    subset(action == "searchResultPage", select = c("date_week", "n_results"))
```

```
bootstraps_q3 <- boot(data = events_q3ba,
                      statistic = funcao_bootstrap_q3_global, # <- referência para a função
                      R = 1000) # número de bootstraps

glimpse(bootstraps_q3$t)
```

```
##  num [1:1000, 1] 18.5 18.4 18.3 18.5 18.5 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q3$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
```

```
## boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (18.27, 18.81 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a taxa de pesquisas sem resultado no grupo A da população está entre 18,28% e 18,82%.

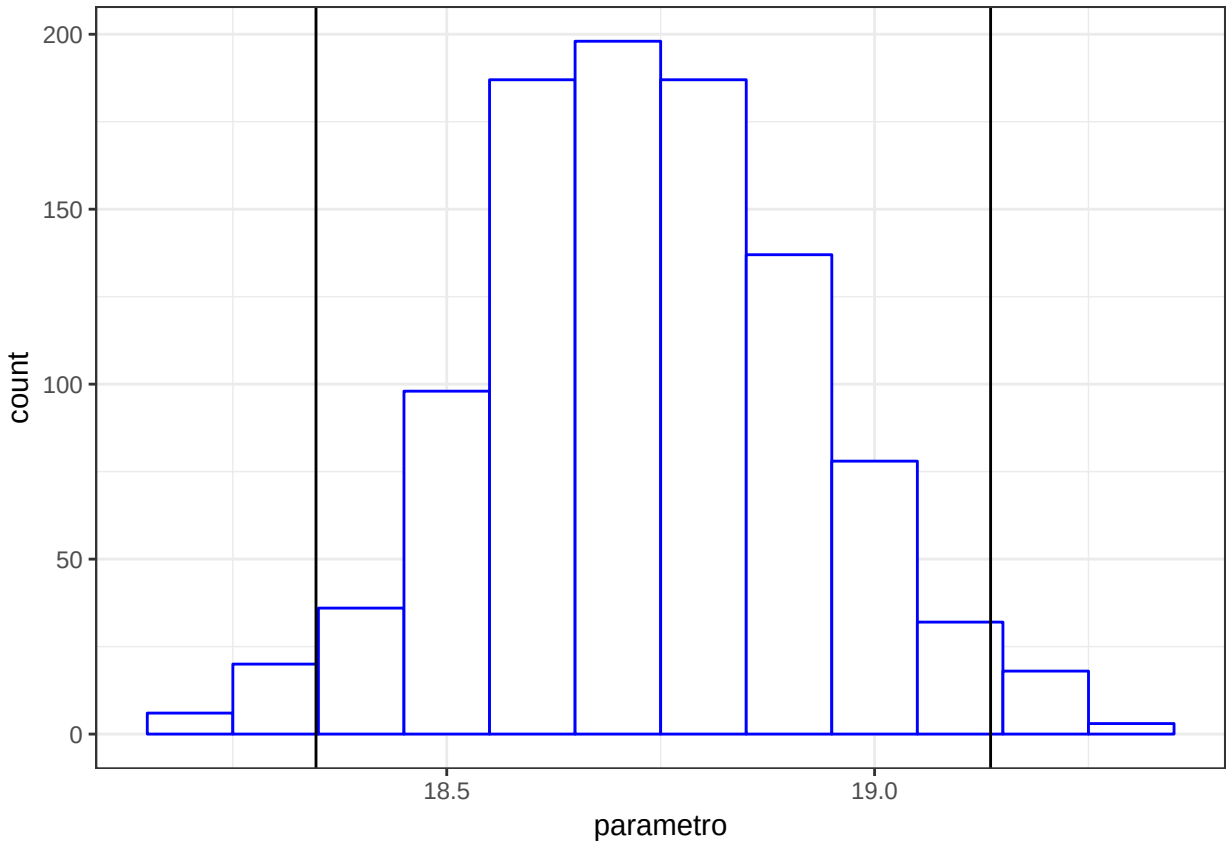
Analisando a média de taxa de pesquisas sem resultado diária geral (do grupo B)

```
events_q3bb = events %>%
    filter(group == "b") %>%
    group_by(date_week, session_id, search_index) %>%
    subset(action == "searchResultPage", select = c("date_week", "n_results"))


bootstraps_q3 <- boot(data = events_q3bb,
                  statistic = funcao_bootstrap_q3_global, # <- referência para a função
                  R = 1000) # número de bootstraps

glimpse(bootstraps_q3$t)
```

```
##  num [1:1000, 1] 18.8 18.6 18.9 18.7 18.8 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```
tibble(parametro = as.double(bootstraps_q3$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (18.33, 19.11 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a taxa de pesquisas sem resultado no grupo B da população está entre 18,35% e 19,12%.

Analisando a diferença da média de taxa de pesquisas sem resultado diária entre o grupo A e o grupo B

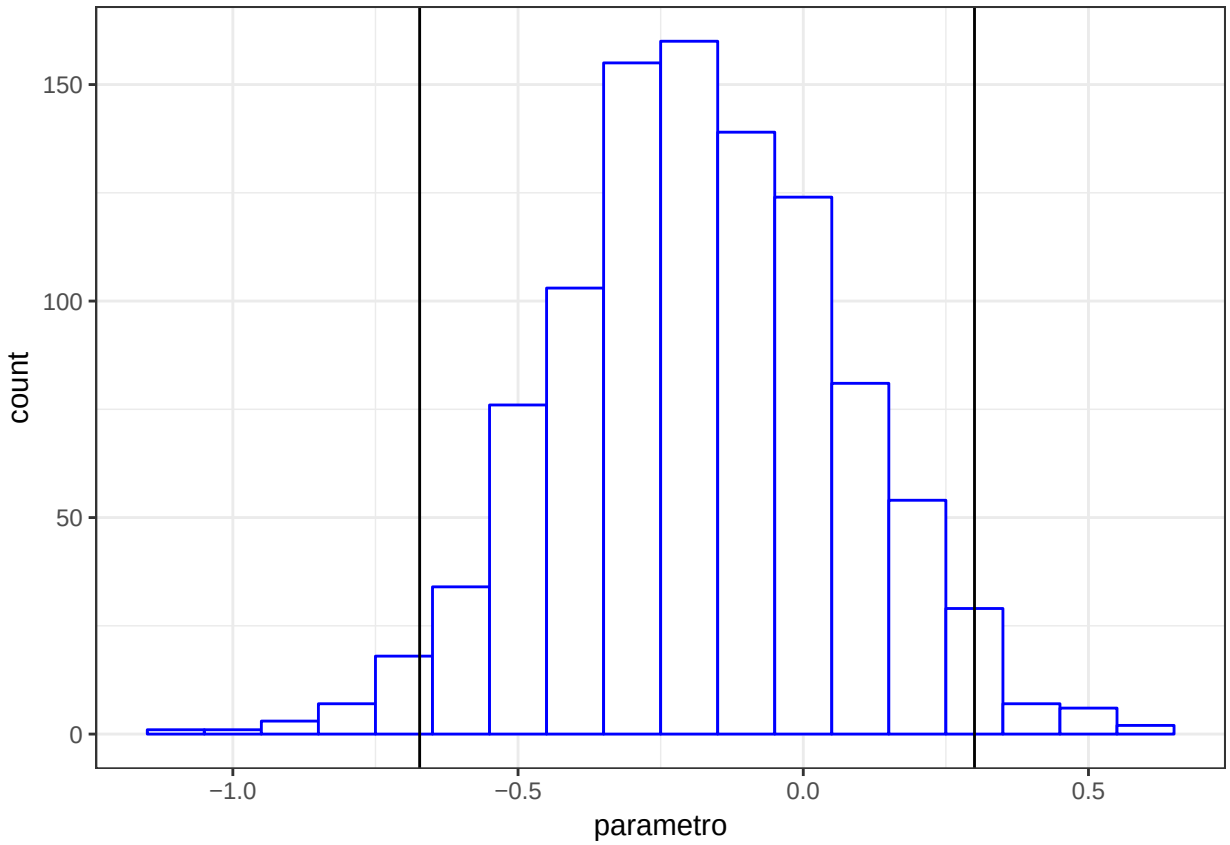
```
events_q3bdif = events %>%
    group_by(group, date_week, session_id, search_index) %>%
    subset(action == "searchResultPage", select = c("group", "date_week", "n_results"))
```

```r
funcao_bootstrap_q3_dif <- function(data, indexes){
  daily_rate = data %>%
    slice(indexes) %>%
    group_by(group, date_week) %>%
    summarize(
      zero_rate = (sum(n_results==0) / sum(n_results>=0)) * 100
    )

  mean_rate_a = daily_rate %>%
    filter(group == "a") %>%
    summarize(
      mean_rate = mean(zero_rate)
    ) %>%
    pull (mean_rate)

  mean_rate_b = daily_rate %>%
    filter(group == "b") %>%
    summarize(
      mean_rate = mean(zero_rate)
    ) %>%
    pull (mean_rate)

  return(
    mean(mean_rate_a - mean_rate_b)
  )
}

bootstraps_q3 <- boot(data = events_q3bdif,
                      statistic = funcao_bootstrap_q3_dif, # <- referência para a função
                      R = 1000) # número de bootstraps

glimpse(bootstraps_q3$t)
```

```
##  num [1:1000, 1] 0.153 -0.1416 0.223 -0.0843 0.1663 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```r
tibble(parametro = as.double(bootstraps_q3$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q3$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```
boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q3, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (-0.6766,  0.3191 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a média da taxa de pesquisas sem resultado diária no grupo A não é significativamente distinguível em relação à mesma medida do grupo B, tendo em vista que a diferença oscila entre -0.6554 ponto percentual (observar que trata-se de um limite inferior negativo) e +0,2904 ponto percentual (positivo). Portanto, a hipótese de desse valores serem iguais nos dois grupos é plausível e não pode ser descartada.

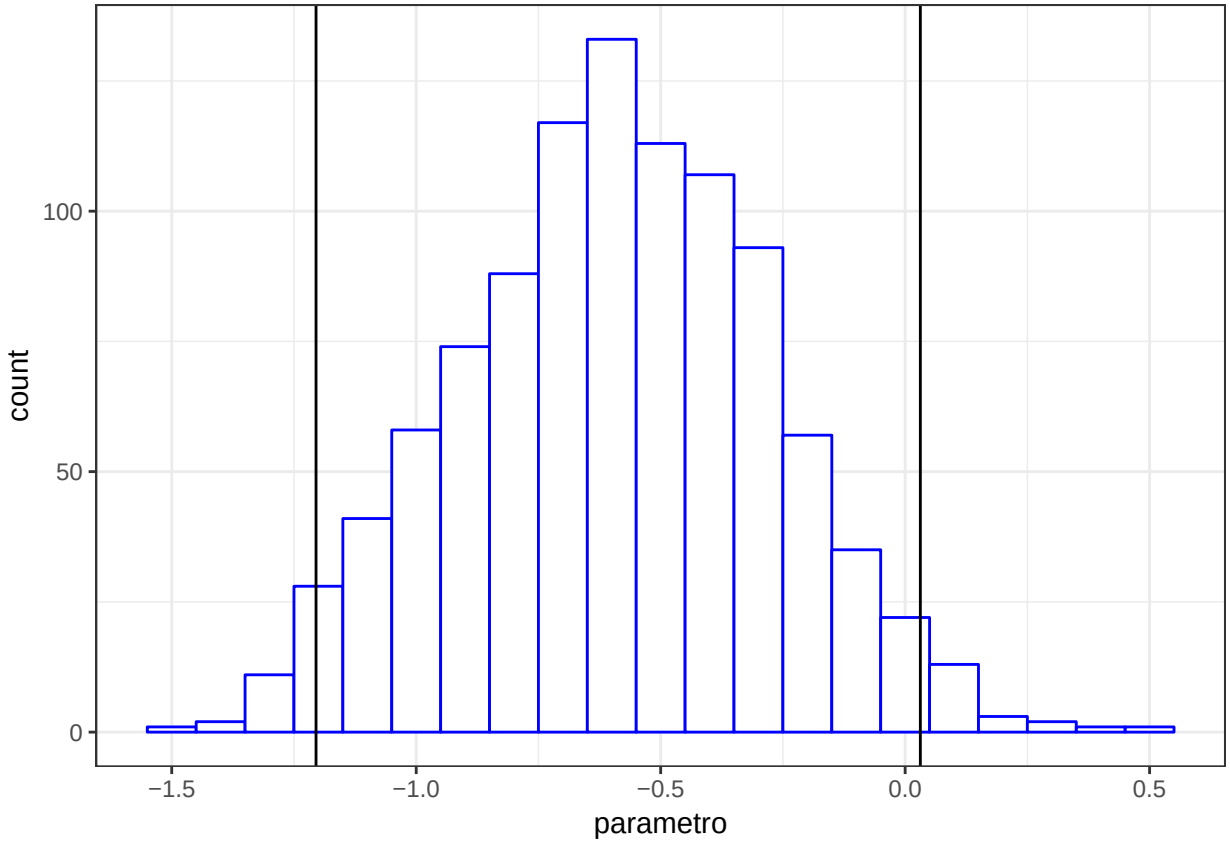
Criando uma comparação A/A na pergunta 1

```r
events_q1aa = events %>%
    group_by(group, date_week, session_id, search_index) %>%
    mutate(click_count = max(visit_index)) %>%
    subset(action == "searchResultPage" & group == "a", select = c("group", "date_week", "click_count"))

#Gerando os índices da amostra A0 e da amostra A1

  indexes_a = vector(mode = "integer", nrow(events_q1aa))
  indexes_a[sample(1:nrow(events_q1aa), size = trunc(nrow(events_q1aa)/2), replace = FALSE)] = 1
  indexes_a0 = which(indexes_a == 0)
  indexes_a1 = which(indexes_a == 1)

#Renomeando o grupo A para virar A0 e A1
events_q1aa$group[indexes_a0] = "a0"
events_q1aa$group[indexes_a1] = "a1"

funcao_bootstrap_q1_aa <- function(data, indexes){
  daily_rate = data %>%
    slice(indexes) %>%
    group_by(group, date_week) %>%
    summarize(
      clickthrough_rate = (sum(click_count>0) / sum(click_count>=0)) * 100
    )
  mean_rate_a0 = daily_rate %>%
    filter(group == "a0") %>%
    summarize(
      mean_rate = mean(clickthrough_rate)
    ) %>%
    pull (mean_rate)

  mean_rate_a1 = daily_rate %>%
    filter(group == "a1") %>%
    summarize(
      mean_rate = mean(clickthrough_rate)
    ) %>%
    pull (mean_rate)

  return(
    mean(mean_rate_a0 - mean_rate_a1)
  )
}

bootstraps_q1aa <- boot(data = events_q1aa,
                  statistic = funcao_bootstrap_q1_aa, # <- referência para a função
                  R = 1000) # número de bootstraps

glimpse(bootstraps_q1aa$t)
```

```
##  num [1:1000, 1] -0.744 -0.576 -0.716 -0.758 -0.462 ...
```

Vamos ver como ficou a distribuição amostral do parâmetro populacional estimado via bootstrap. Estão marcadas no histograma duas linhas verticais para os quantis 2,5% e 97,5%.

```r
tibble(parametro = as.double(bootstraps_q1aa$t)) %>%
  ggplot(aes(x = parametro)) +
  geom_histogram(binwidth = 0.1, fill = "white", color = "blue")+
  geom_vline(xintercept = quantile(bootstraps_q1aa$t, 0.025)[[1]]) +
  geom_vline(xintercept = quantile(bootstraps_q1aa$t, 0.975)[[1]])
```



Agora sim, vamos solicitar o cálculo dos intervalos de confiança que contém o parâmetro calculado em 95% das vezes que esse cálculo for realizado.

```r
boot.ci(boot.out = bootstraps_q1aa, conf = 0.95, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstraps_q1aa, conf = 0.95, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (-1.1725,  0.0669 )
## Calculations and Intervals on Original Scale
```

Portanto, baseado nesse bootstrap, podemos afirmar com 95% de confiança que a média da taxa de cliques diária dentro do próprio grupo A não apresenta diferença significativa.