DN2. DNS: longevidade de nomes

Rafilx

2022-05-02

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: viridisLite

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

R Markdown

A longevidade de um nome (QNAME+QTYPE) no dataset pode ser definida como o intervalo entre a primeira e a última aparição desse nome. Calcular a longevidade dos nomes no dataset, e analisar como essa variável está distribuída.

Resultados esperados:

- análise gráfica da distribuição (histograma, ECDF) e numérica (min, max, média, mediana) da longevidade dos nomes
  - por enquanto não vejo sentido em dividir a análise por período, então pode considerar o dataset como um todo
  - minha intuição é que a distribuição seja assimétrica com (longa) cauda à direita
- Busca os dados no banco com o parse do DNS ja realizado, então temos:
  - qname que é o domínio
  - QTYPE tipo da query
  - query_id ID da transação definido pelo atacante
  - year_period ano e trimestre em que ocorreu o ataque exemplo "20212" o ataque ocorreu no segundo trimestre do 2021

```
db <- dbConnect(RSQLite::SQLite(), dbname="../dnstor_statistics_dns.sqlite")

data_unfetch <-dbSendQuery(db, "
  SELECT *
    FROM DNS_ANALYSIS
        JOIN DNS_ANALYSIS_QUESTION
          ON DNS_ANALYSIS.id = DNS_ANALYSIS_QUESTION.dns_analysis_id
     WHERE QTYPE != 0
")
data <- fetch(data_unfetch)

dbDisconnect(db)
```

```
## Warning in connection_release(conn@ptr): There are 1 result in use. The
## connection will be released when they are closed
```

```
data['tempo_final_cast'] = as.POSIXct(data[['tempo_final']], format = "%Y-%m-%d %H:%M:%S")
data['tempo_inicio_cast'] = as.POSIXct(data[['tempo_inicio']], format = "%Y-%m-%d %H:%M:%S")

data_grouped = data %>%
  group_by(qname, qtype) %>%
  summarise(tempo_inicio=min(tempo_inicio_cast), tempo_final=max(tempo_final_cast)) %>%
  mutate(tempo_diff = tempo_final - tempo_inicio) %>%
  #filter(tempo_diff > 0) %>%
  arrange(desc(tempo_diff))
```

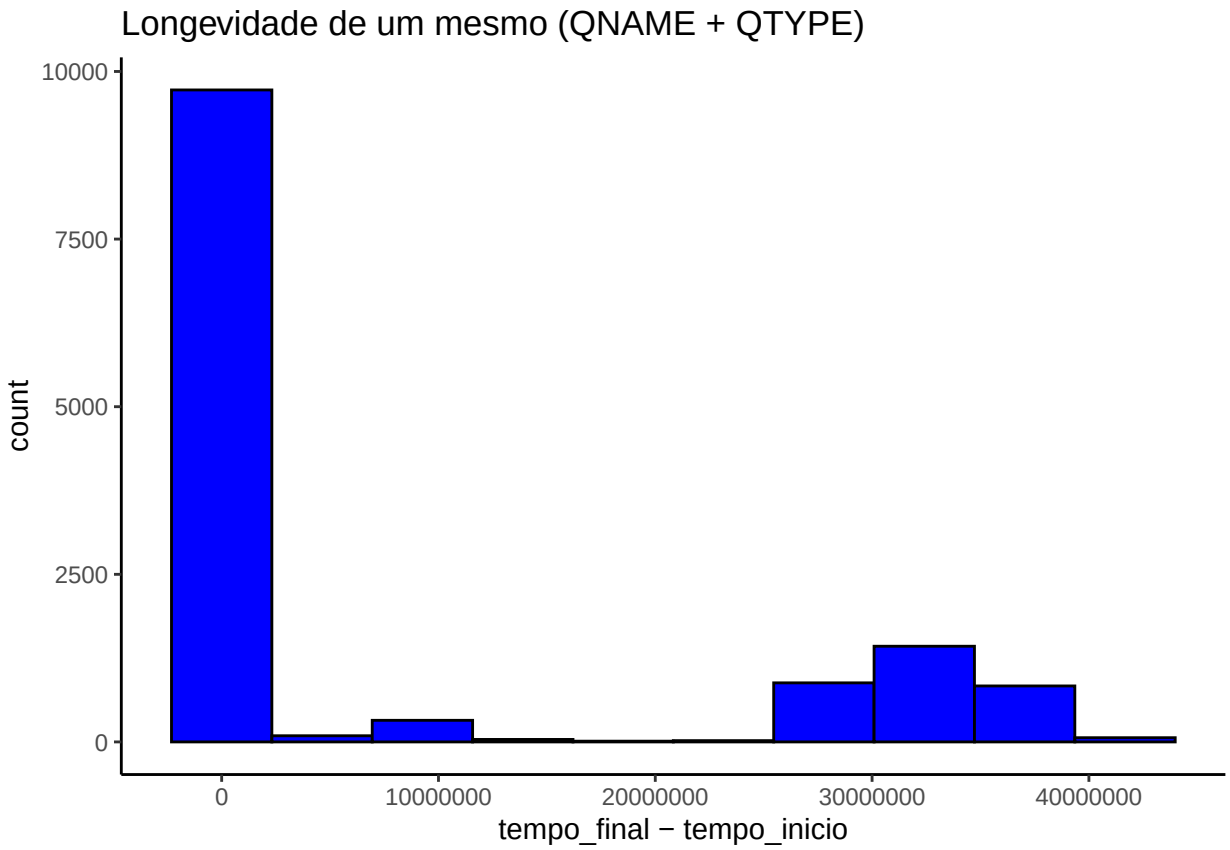
```
## `summarise()` has grouped output by 'qname'. You can override using the
## `.groups` argument.
```

```
data_grouped %>%
  head(10)
```

```
## # A tibble: 10 x 5
## # Groups:   qname [10]
##    qname                qtype tempo_inicio        tempo_final         tempo_diff
##    <chr>                <chr> <dttm>              <dttm>              <drtn>
##  1 VERSION.BIND.        TXT   2020-10-30 02:39:27 2022-02-24 08:33:01 41666014 ~
##  2 whoami.akamai.net.   A     2020-10-30 12:33:16 2022-02-23 10:35:19 41551323 ~
##  3 com.                 ANY   2020-10-31 11:38:32 2022-02-24 09:16:50 41549898 ~
##  4 version.bind.        TXT   2020-11-01 04:46:10 2022-02-24 15:42:17 41511367 ~
##  5 isc.org.             ANY   2020-11-01 22:40:23 2022-02-24 02:25:52 41399129 ~
##  6 researchscan541.eec~ A     2020-11-01 15:23:29 2022-02-23 14:06:13 41380964 ~
##  7 adsports.ae.         MX    2020-11-03 10:59:02 2022-02-23 23:36:50 41258268 ~
##  8 public1.114dns.com.  A     2020-11-04 09:47:54 2022-02-24 15:42:14 41234060 ~
##  9 238.107.19.200.in-a~ PTR   2020-11-01 23:58:38 2022-02-21 13:45:36 41176018 ~
## 10 a.gtld-servers.net.  A     2020-10-30 02:55:07 2022-02-18 00:20:55 41117148 ~
```

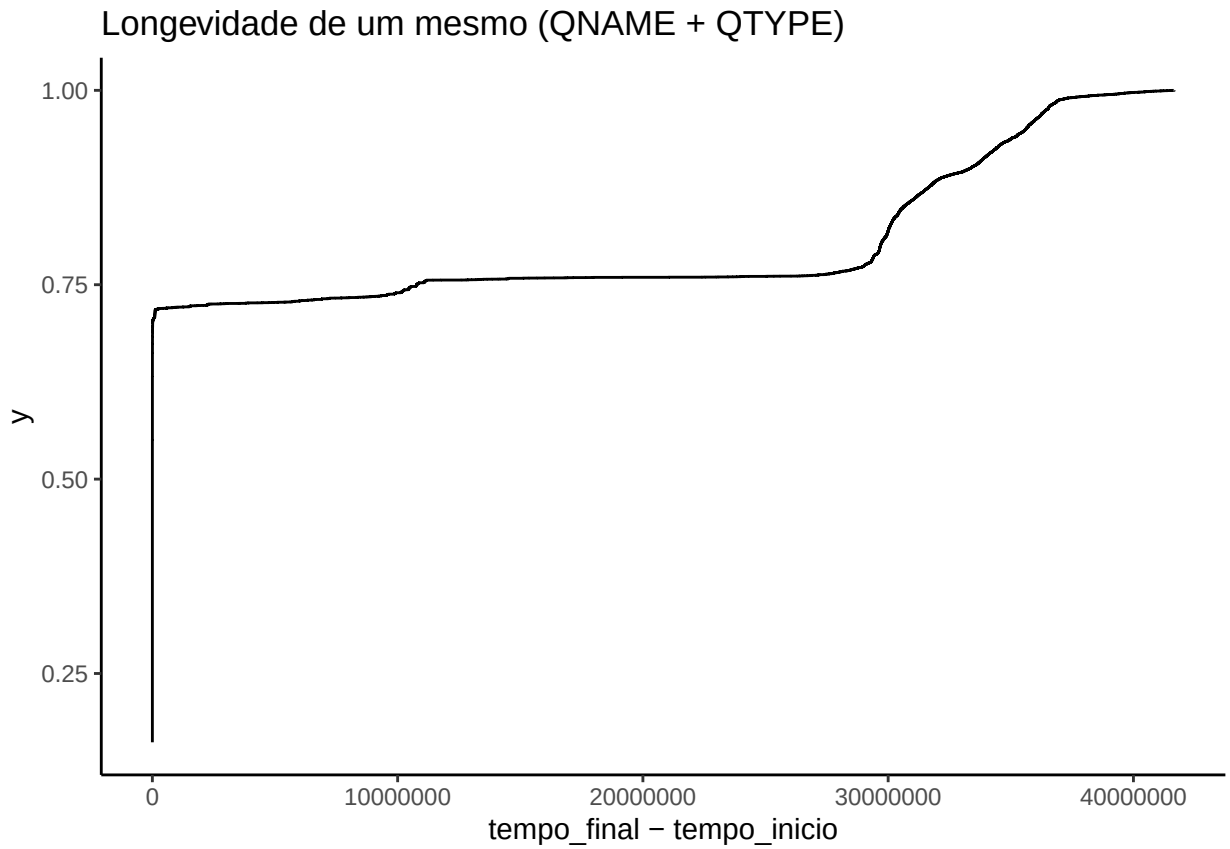
```
data_grouped %>%
  ggplot(aes(x= tempo_diff)) +
  geom_histogram(bins = 10, fill='blue', color ='black') +
  ggtitle("Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("tempo_final - tempo_inicio") +
  theme_classic()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
data_grouped %>%
  ggplot(aes(x= tempo_diff)) +
  stat_ecdf(geom = "step", pad = FALSE) +
  ggtitle("Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("tempo_final - tempo_inicio") +
  theme_classic()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
data_grouped.tempo_diff.min = min(data_grouped$tempo_diff)
data_grouped.tempo_diff.max = max(data_grouped$tempo_diff)
data_grouped.tempo_diff.mean = mean(data_grouped$tempo_diff)
data_grouped.tempo_diff.median = median(data_grouped$tempo_diff)

quantile(data_grouped$tempo_diff)
```

```
## Time differences in secs
##       0%      25%      50%      75%     100%
##        0       14       69 10780940 41666014
```

```
summary(data_grouped)
```

```
##     qname              qtype            tempo_inicio
##  Length:13411       Length:13411       Min.   :2020-10-29 16:15:05
##  Class :character   Class :character   1st Qu.:2020-12-13 09:03:49
##  Mode  :character   Mode  :character   Median :2021-01-01 02:01:27
##                                        Mean   :2021-04-27 18:46:19
##                                        3rd Qu.:2021-10-25 18:19:09
##                                        Max.   :2022-02-24 15:42:16
##   tempo_final                    tempo_diff
##  Min.   :2020-10-29 23:17:13   Length:13411
##  1st Qu.:2020-12-15 00:54:54   Class :difftime
##  Median :2021-10-25 06:53:39   Mode  :numeric
```

```
##  Mean   :2021-07-31 07:23:43
##  3rd Qu.:2021-12-06 22:50:36
##  Max.   :2022-02-24 15:49:24
```

- Dados sobre o intervalo entre a primeira e a última aparição desse (QNAME+QTYPE)
  - Mínimo 0 segundos
  - Máximo 41666014 segundos
  - Média 8167043.84 segundos
  - Mediana 69 segundos