

DN2. DNS: longevidade de nomes

Rafilx

2022-08-13

```
##
## 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 já 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="../db/database-2022-05-11/dnstor_statistics_dns.sqlite")

data_unfetch <-dbSendQuery(db, "
  SELECT *, CAST(CAST(year AS text) || CAST(period AS text) as integer) as year_period
  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")

secs_to_hours = (60 * 60)
secs_to_days = (secs_to_hours * 24)

data_grouped = data %>%
  group_by(qname, qtype) %>%
  summarise(tempo_inicio=min(tempo_inicio_cast), tempo_final=max(tempo_final_cast),
    sum_requests_per_attack=sum(requests_per_attack), number_of_attacks=n()) %>%
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"),
    tempo_diff = tempo_final - tempo_inicio,
    avg_requests_per_attack=sum_requests_per_attack/number_of_attacks) %>%
  arrange(desc(tempo_diff_secs))
```

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

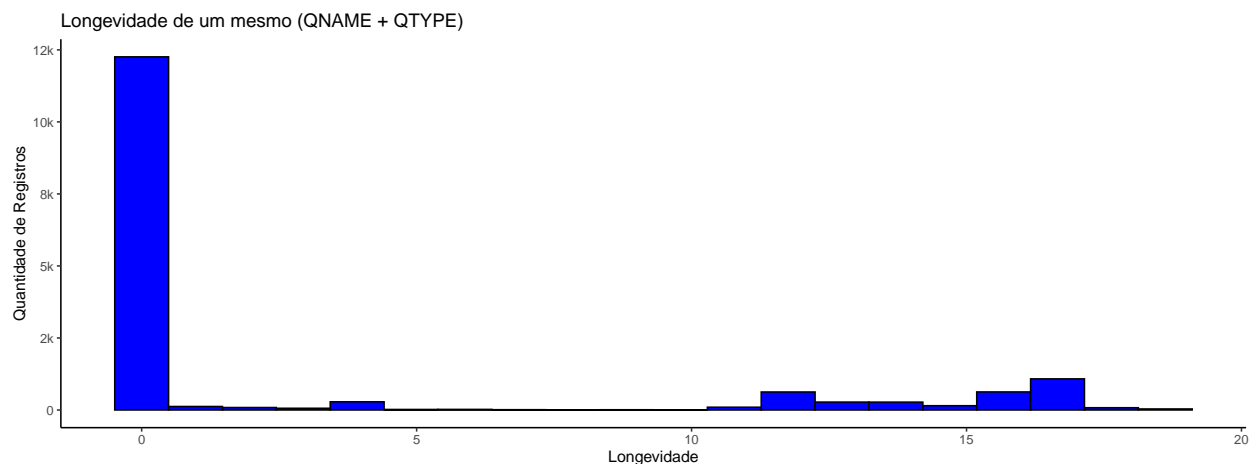
```
data_grouped %>%
  head(10)
```

```
## # A tibble: 10 x 9
## # Groups:   qname [10]
##   qname    qtype tempo_inicio      tempo_final      sum_r~1 numbe~2 tempo~3
##   <chr>   <chr> <dtm>          <dtm>          <int>   <int>   <dbl>
## 1 VERSIO~ TXT   2020-10-30 02:39:27 2022-05-11 13:05:22 6.16e3    2531  4.82e7
## 2 peacece~ ANY   2020-10-31 14:28:23 2022-05-11 05:22:49 7.27e7  118333  4.81e7
## 3 200-19~ A     2020-10-30 11:15:36 2022-05-09 21:23:22 1.48e2     98  4.81e7
## 4 versio~ TXT   2020-11-01 04:46:10 2022-05-11 12:13:52 8.33e2    665  4.81e7
## 5 a.gtld~ A     2020-10-30 02:55:07 2022-05-09 06:50:12 8.9 e1     84  4.81e7
## 6 com.    ANY   2020-10-31 11:38:32 2022-05-09 16:38:23 5.11e2    428  4.80e7
## 7 238.10~ PTR   2020-11-01 23:58:38 2022-05-09 04:56:13 8.42e2     32  4.78e7
## 8 dns-te~ TXT   2020-11-04 06:49:48 2022-05-11 09:58:47 8.7 e1     85  4.78e7
## 9 mopa.a~ MX    2020-11-04 14:39:23 2022-05-11 13:52:05 6.10e3     76  4.78e7
## 10 szgmc.~ MX    2020-11-03 14:47:10 2022-05-10 13:29:54 8.89e2     63  4.78e7
## # ... with 2 more variables: tempo_diff <drtn>, avg_requests_per_attack <dbl>,
```

```
## # and abbreviated variable names 1: sum_requests_per_attack,
## # 2: number_of_attacks, 3: tempo_diff_secs
```

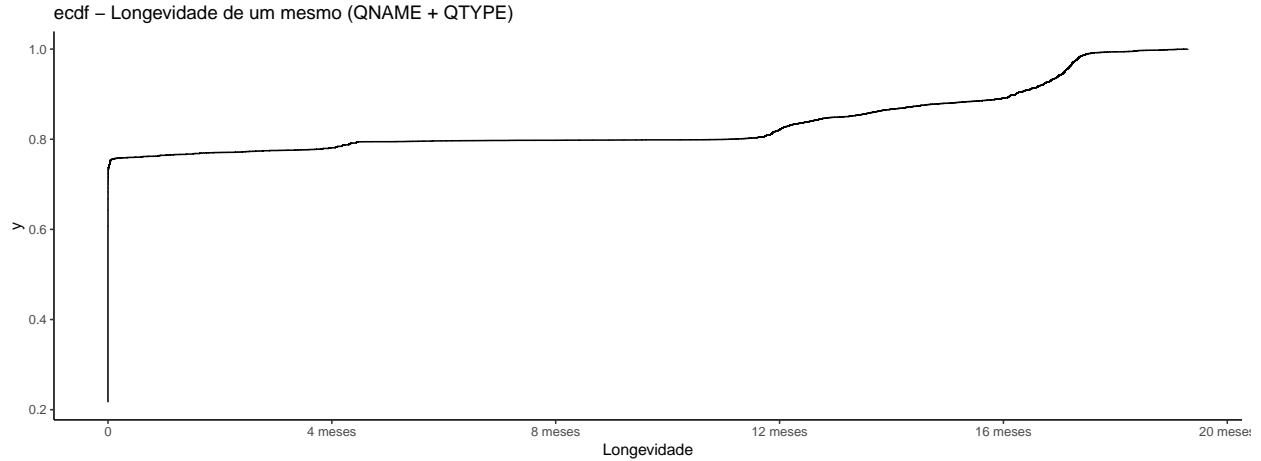
- Dessa forma é buscado apresentar quanto tempo um QNAME em conjunto com o QTYPE, permanece ativo, ex o qname “peacecorps.gov.” do qtype “ANY” teve o primeiro ataque registrado em 31/10/2020 e o ultimo ataque com o mesmo qname e qtype em 11/05/2022 totalizando uma longevidade de aproximadamente um ano e meio (48092066 secs)

```
data_grouped %>%
  mutate(tempo_diff_secs = tempo_diff_secs / months) %>%
  ggplot(aes(x= tempo_diff_secs)) +
  geom_histogram(bins = 20, fill='blue', color = 'black') +
  ggtitle("Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("Longevidade") +
  ylab("Quantidade de Registros") +
  scale_y_continuous(labels = addUnits) +
  # scale_x_continuous(labels = addUnitsHours) +
  # scale_x_continuous(labels = longevidadeUnits) +
  # scale_x_discrete(limits = c(0, 16*months), breaks = 0:(4*months)) +
  # xlim("0", "10000000", "20000000", "30000000", "40000000") +
  # scale_x_discrete(
  #   breaks=c("0", "10000000", "20000000", "30000000", "40000000"),
  #   limits = c("0", "4 meses", "8 meses", "12 meses", "16 meses")) +
  #   limits = c("0" = "0", "10000000" = "4 meses", "20000000" = "8 meses", "30000000" = "12 meses", "40000000" = "16 meses")
  theme_classic()
```



```
# verificar para colocar a cada 3 ou 4 meses
```

```
data_grouped %>%
  ggplot(aes(x= tempo_diff_secs)) +
  stat_ecdf(geom = "step", pad = FALSE) +
  ggtitle("ecdf - Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("Longevidade") +
  scale_x_continuous(labels = longevidadeUnits) +
  theme_classic()
```



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

quantile(data_grouped$tempo_diff_secs)
```

```
##      0%      25%      50%      75%     100%
##      0       5      67    105178 48248755
```

```
summary(data_grouped)
```

```
##      qname          qtype      tempo_inicio
## Length:16125      Length:16125      Min.   :2020-10-29 16:15:05.00
## Class :character  Class :character  1st Qu.:2020-12-17 16:50:24.00
## Mode  :character  Mode  :character  Median :2021-07-09 00:38:37.00
##                                     Mean  :2021-06-25 05:28:30.13
##                                     3rd Qu.:2021-11-15 13:56:59.00
##                                     Max.   :2022-05-11 13:22:23.00
##      tempo_final      sum_requests_per_attack number_of_attacks
## Min.   :2020-10-29 23:17:13.00      Min.   :      1      Min.   :      1
## 1st Qu.:2021-03-08 14:18:01.00      1st Qu.:      2      1st Qu.:      1
## Median :2021-11-01 10:55:05.00      Median :      9      Median :      1
## Mean   :2021-09-25 18:50:31.26      Mean   :    5580      Mean   :     36
## 3rd Qu.:2022-04-02 11:48:10.00      3rd Qu.:    101      3rd Qu.:      4
## Max.   :2022-05-11 14:41:27.00      Max.   :72734156      Max.   :118333
##      tempo_diff_secs      tempo_diff      avg_requests_per_attack
## Min.   :      0      Length:16125      Min.   :      1
## 1st Qu.:      5      Class :difftime  1st Qu.:      2
## Median :     67      Mode  :numeric  Median :      8
## Mean   : 7996921                      Mean   :     25
## 3rd Qu.: 105178                      3rd Qu.:     21
## Max.   :48248755                      Max.   :18556
```

- Dados sobre o intervalo entre a primeira e a última aparição desse (QNAME+QTYPE)
 - Mínimo 0 segundos

- Máximo 558.4347 meses
- Média 133282.019 minutos
- Mediana 67 segundos

```
trim_value = .1

data_grouped$tempo_diff_secs.min = min(data_grouped$tempo_diff_secs, trim=trim_value)
data_grouped$tempo_diff_secs.max = max(data_grouped$tempo_diff_secs, trim=trim_value)
data_grouped$tempo_diff_secs.mean = mean(data_grouped$tempo_diff_secs, trim=trim_value)
data_grouped$tempo_diff_secs.median = median(data_grouped$tempo_diff_secs, trim=trim_value)

quantile(data_grouped$tempo_diff_secs, trim=trim_value)
```

```
##      0%      25%      50%      75%     100%
##      0       5      67    105178 48248755
```

```
summary(data_grouped, trim=trim_value)
```

```
##      qname              qtype      tempo_inicio
## Length:16125      Length:16125      Min.   :2020-10-29 16:15:05.00
## Class :character  Class :character  1st Qu.:2020-12-17 16:50:24.00
## Mode  :character  Mode  :character  Median :2021-07-09 00:38:37.00
##                                     Mean  :2021-06-25 05:28:30.13
##                                     3rd Qu.:2021-11-15 13:56:59.00
##                                     Max.   :2022-05-11 13:22:23.00
##      tempo_final      sum_requests_per_attack number_of_attacks
## Min.   :2020-10-29 23:17:13.00      Min.   :      1      Min.   :      1
## 1st Qu.:2021-03-08 14:18:01.00      1st Qu.:      2      1st Qu.:      1
## Median :2021-11-01 10:55:05.00      Median :      9      Median :      1
## Mean   :2021-09-25 18:50:31.26      Mean   :    5580      Mean   :     36
## 3rd Qu.:2022-04-02 11:48:10.00      3rd Qu.:    101      3rd Qu.:      4
## Max.   :2022-05-11 14:41:27.00      Max.   :72734156      Max.   :118333
##      tempo_diff_secs      tempo_diff      avg_requests_per_attack
## Min.   :      0      Length:16125      Min.   :      1
## 1st Qu.:      5      Class :difftime  1st Qu.:      2
## Median :     67      Mode  :numeric  Median :      8
## Mean   : 7996921                      Mean   :     25
## 3rd Qu.: 105178                      3rd Qu.:     21
## Max.   :48248755                      Max.   :18556
```

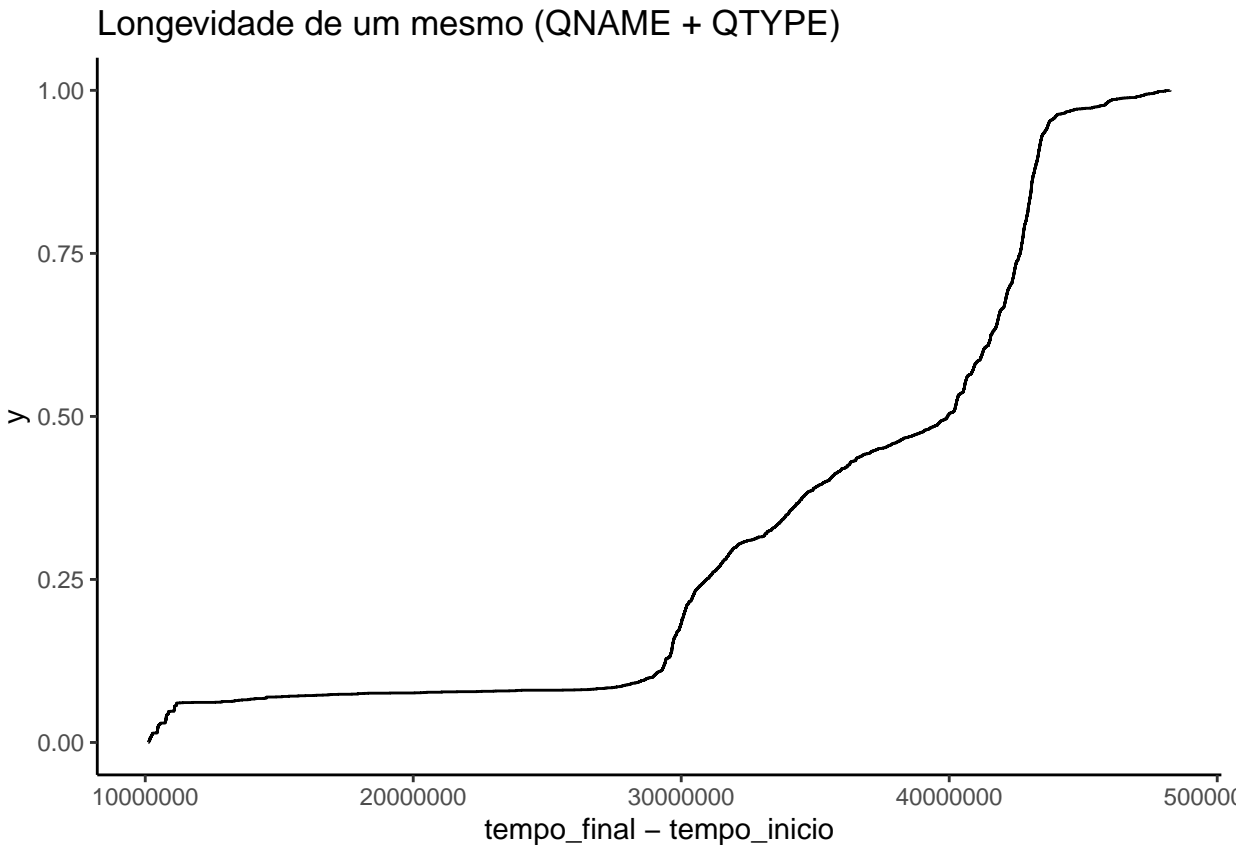
- Dados sobre o intervalo entre a primeira e a última aparição do (QNAME+QTYPE) removendo 10% dos valores máximos e mínimos
 - Mínimo 0 segundos
 - Máximo 558.4347 meses
 - Média 77605.1631 minutos
 - Mediana 67 segundos
- Identificar qual é o percentil em que há essa mudança de tendência (próximo aos 72-73%) e qual a duração correspondente

```
quantile(data_grouped$tempo_diff_secs,
         c(.5332, .6235, .696282, .710307, .72, .7204,
           .721090026, .73, .7396902445, .75, .76,
           .77, .78, .99))
```

```
##      53.32%      62.35%      69.6282%      71.0307%      72%      72.04%      72.109%
##      70.0      745.6      3602.7      5457.6      7265.0      7434.6      7764.7
##      73%      73.96902%      75%      76%      77%      78%      99%
##     12521.6     41400.8     105178.0    1229816.9    4526143.5    9874846.8    43799901.3
```

- Isso representa que Y% dos (QNAME+TYPE), tem os seus ataques com duração de até X segundos:
 - 53.3% até 100 segundos
 - 62.3% até 1000 segundos
 - 69.6% até 10000 segundos
 - 71% até 100000 segundos
 - 72% até 598289 segundos
 - 72.04% até 803076 segundos
 - 72.11% até 1000000 segundos
 - 73.97% até 10000000 segundos
- Significa que após os (69%) o tempo dos ataques cresceu de 27.7778 até 115.7407 (73.97%) onde estabiliza próximo dos 10000000 segundos
- Uma representação ECDF removendo os registros abaixo da quantidade de segundos em que apresenta estabilidade (10000000 segundos)

```
data_grouped %>%
  filter(tempo_diff_secs > 10000000) %>%
  ggplot(aes(x= tempo_diff_secs)) +
  stat_ecdf(geom = "step", pad = FALSE) +
  ggtitle("Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("tempo_final - tempo_inicio") +
  theme_classic()
```



```
percentage_76_secs = quantile(data_grouped$tempo_diff_secs, c(.76))[[1]]
percentage_76_secs/secs_to_days
```

```
## [1] 14.23
```

```
percentage_99_secs = quantile(data_grouped$tempo_diff_secs, c(.995))[[1]]
percentage_99_secs/secs_to_days
```

```
## [1] 529.7
```

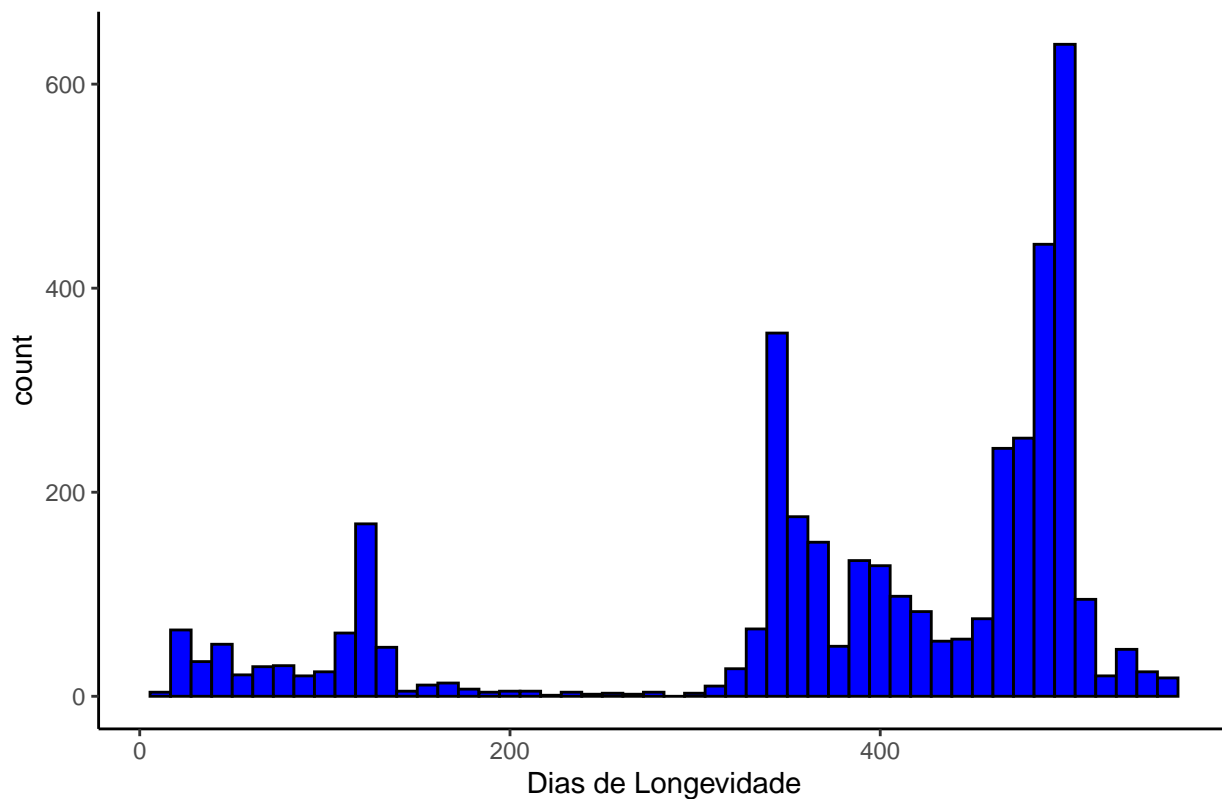
```
data_grouped$tempo_diff_secs.max / secs_to_days
```

```
## [1] 558.4
```

- Cerca de 24% dos (QNAME + QTYPE) possuem uma longevidade entre 9 e 16 meses

```
data_grouped %>%
  filter(tempo_diff_secs > percentage_76_secs) %>%
  ggplot(aes(x= tempo_diff_secs / secs_to_days)) +
  geom_histogram(bins = 50, fill='blue', color='black') +
  ggtitle("Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("Dias de Longevidade") +
  theme_classic()
```

Longevidade de um mesmo (QNAME + QTYPE)



```
data_bigger_than_76 = data_grouped %>%
  filter(tempo_diff_secs > percentage_76_secs) %>%
  ungroup() %>%
  group_by(qtype) %>%
  summarise(qtype_quantity = n()) %>%
  arrange(desc(qtype_quantity))

sum_qtype_quantity = sum(data_bigger_than_76$qtype_quantity)
data_bigger_than_76_percentage = data_bigger_than_76 %>%
  mutate(qtype_quantity_percentage = (qtype_quantity / sum_qtype_quantity) * 100)

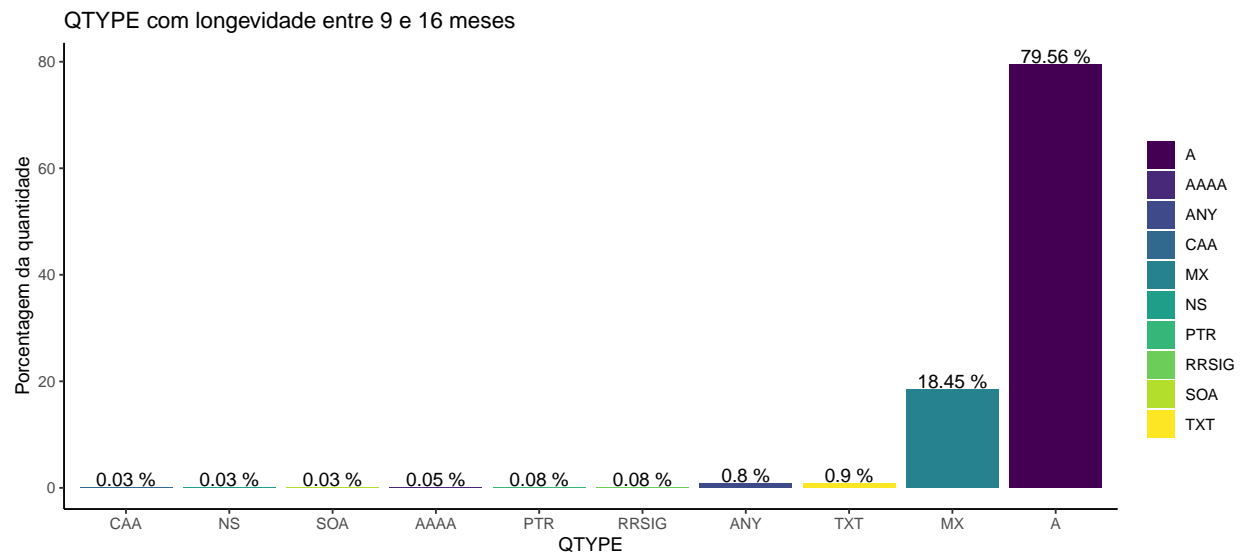
data_bigger_than_76_percentage
```

```
## # A tibble: 10 x 3
##   qtype qtype_quantity qtype_quantity_percentage
##   <chr>         <int>             <dbl>
## 1 A             3079             79.6
## 2 MX             714             18.4
## 3 TXT             35              0.904
## 4 ANY             31              0.801
## 5 PTR              3              0.0775
## 6 RRSIG            3              0.0775
## 7 AAAA             2              0.0517
## 8 CAA              1              0.0258
```



```
## 9 NS 1 0.0258
## 10 SOA 1 0.0258
```

```
data_bigger_than_76_percentage %>%
  ggplot( aes(x=reorder(qtype, +qtype_quantity_percentage), y=qtype_quantity_percentage, fill=qtype)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_viridis(discrete=TRUE, name="") +
  geom_text(aes(label = paste(round(qtype_quantity_percentage, 2), "%")), vjust = -0.10, ) +
  theme_classic() +
  ylab("Porcentagem da quantidade") +
  xlab("QTYPE") +
  ggtitle("QTYPE com longevidade entre 9 e 16 meses")
```



- Então dos QTYPE que possuem uma alta longevidade entre 9 e 16 meses (cerca de 24% de todos os registros 76% ~ 100%)
 - 20% (714) deles possuem o QTYPE “MX”
 - 78% (3079) dos ataques com maior longevidade utilizam o QTYPE “A”, o que é surpreendente
 - E por fim o QTYPE “ANY” aparece com apenas 31 registros de QTYPE com longevidade entre 9 e 16 meses

```
percentage_76_secs_A_qnames = data_grouped %>%
  filter(tempo_diff_secs > percentage_76_secs) %>%
  filter(qtype == "A") %>%
  select(qname) %>%
  distinct(qname)

percentage_76_secs_qtype_A = data %>%
  filter(qtype == "A") %>%
  filter(qname %in% percentage_76_secs_A_qnames$qname)
```

- O QTYPE “A” é o QTYPE que possui a maior quantidade de QNAMEs com alta longevidade entre 9 e 16 meses

- Esse é o top 10 de QTYPE A agrupado por QNAME representado por qname_count e somado o request_per_attack

```
percentage_76_secs_qtype_A_group_qname = percentage_76_secs_qtype_A %>%
  group_by(qname) %>%
  summarise(qname_count = n(), sum_requests_per_attack=sum(requests_per_attack),
            tempo_inicio=min(tempo_inicio_cast), tempo_final=max(tempo_final_cast),
            sum_requests_per_attack=sum(requests_per_attack)) %>%
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"),
          tempo_diff = tempo_final - tempo_inicio)

percentage_76_secs_qtype_A_group_qname %>%
  arrange(desc(qname_count)) %>%
  select('qname', 'qname_count', 'tempo_diff') %>%
  head(5)
```

```
## # A tibble: 5 x 3
##   qname                                qname_count tempo_diff
##   <chr>                                <int> <drtn>
## 1 whoami.akamai.net.                  5742 543.61 days
## 2 2015annualreport.bloomberg.org.     4264 237.15 days
## 3 www.ndnslab.com.                    3357 130.79 days
## 4 www.ac.my.blastodermic-swimmable.info. 1705 194.58 days
## 5 probe.idealeer.com.                  1108  71.68 days
```

- Unica coisa a ressaltar aqui é que o top 1 QNAME “whoami.akamai.net.” que apareceu 3639x e foi o registro com maior longevidade 480 dias, cerca de 16 meses
- Ao ordenar pela soma de requests por ataque o top 10 muda

```
percentage_76_secs_qtype_A_group_qname %>%
  arrange(desc(sum_requests_per_attack)) %>%
  select('qname', 'qname_count', 'sum_requests_per_attack') %>%
  head(10)
```

```
## # A tibble: 10 x 3
##   qname                                qname_count sum_requests_per_attack
##   <chr>                                <int> <int>
## 1 www.ndnslab.com.                    3357 39546
## 2 www.ac.my.blastodermic-swimmable.info. 1705 28843
## 3 www.ac.my.superability-kooka.info.     771 26210
## 4 admin.asry.net.                      60 21672
## 5 probe.idealeer.com.                  1108 13150
## 6 theguardian.webredirect.org.           62 12805
## 7 ftp.ebisb.com.                       47 12035
## 8 dji.gov.ae.                           63 11492
## 9 hotspot.accesscam.org.                52 11366
## 10 2015annualreport.bloomberg.org.     4264 10967
```

- Nenhum dos registros ordenados pela quantidade de requisições por ataque está no top 10 ordenado pela longevidade dos dados
- Para verificar se o mesmo query_id é muito utilizado foi agrupado somente por query_id

```
percentage_76_secs_qtype_A %>%
  group_by(query_id) %>%
  summarise(query_id_count = n(), sum_requests_per_attack=sum(requests_per_attack), tempo_inicio=min(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_final - tempo_inicio),
  arrange(desc(query_id_count)) %>%
  select('query_id', 'query_id_count', 'sum_requests_per_attack', 'tempo_diff') %>%
  head(10)
```

```
## # A tibble: 10 x 4
##   query_id query_id_count sum_requests_per_attack tempo_diff
##   <int>      <int>          <int> <drtn>
## 1    50265         660          1487 43886164 secs
## 2    45810         323           401 41240463 secs
## 3    28826         215           282 37657765 secs
## 4    64206          99           102 39524383 secs
## 5     4218          98           110 34277498 secs
## 6     1337          86           100 48052505 secs
## 7    44557          76            83 45752078 secs
## 8    14602          64           127 16297096 secs
## 9    16028          49          3793 45945897 secs
## 10   36379          45           142 14703113 secs
```

- Nada chamou a atenção

-
- uma análise mais interessante levando os períodos em comparação é a contagem de nomes novos e retirados (i.e., que deixam de ser usados) por período
 - um nome é contado como novo no período da primeira ocorrência do dataset
 - um nome é contado como retirado no período da última ocorrência no dataset
 - p.ex., um nome que aparece pela primeira vez em 23/04/21 e pela última em 28/10/21 conta como novo em 2021.Q2 e como retirado em 2021.Q4
 - produzir uma tabela e um gráfico de barras (com 2 barras por período, de novos e retirados) com esses dados
 - se a quantidade de nomes novos em 2020.Q4 (1o período) for absurdamente maior que a dos demais períodos, considerar a possibilidade de omiti-la do gráfico

```
data_quarters = data_grouped %>%
  mutate(year_period_inicio=paste(year(tempo_inicio), quarters(tempo_inicio), sep = "."),
         year_period_fim=paste(year(tempo_final), quarters(tempo_final), sep = "."))

data_quarters_new = data_quarters %>%
  ungroup() %>%
  count(year_period_inicio, name = "novos")

data_quarters_removed = data_quarters %>%
  ungroup() %>%
  count(year_period_fim, name = "retirados")

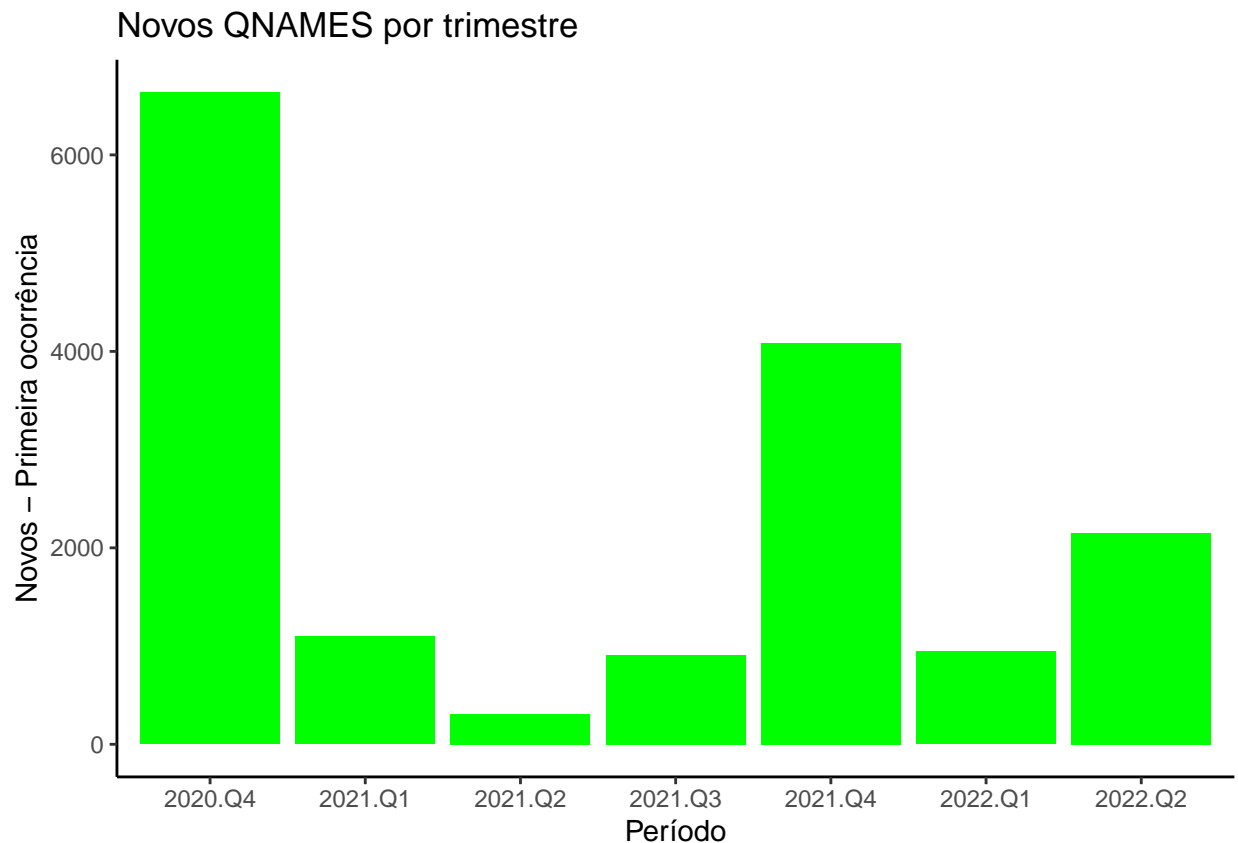
data_quarters_new_removed = data_quarters_new %>%
```

```
inner_join(data_quarters_removed, by = c("year_period_inicio" = "year_period_fim")) %>%
mutate(year_period = as.factor(year_period_inicio) ) %>%
select('year_period', 'novos', 'retirados') %>%
arrange(year_period)
```

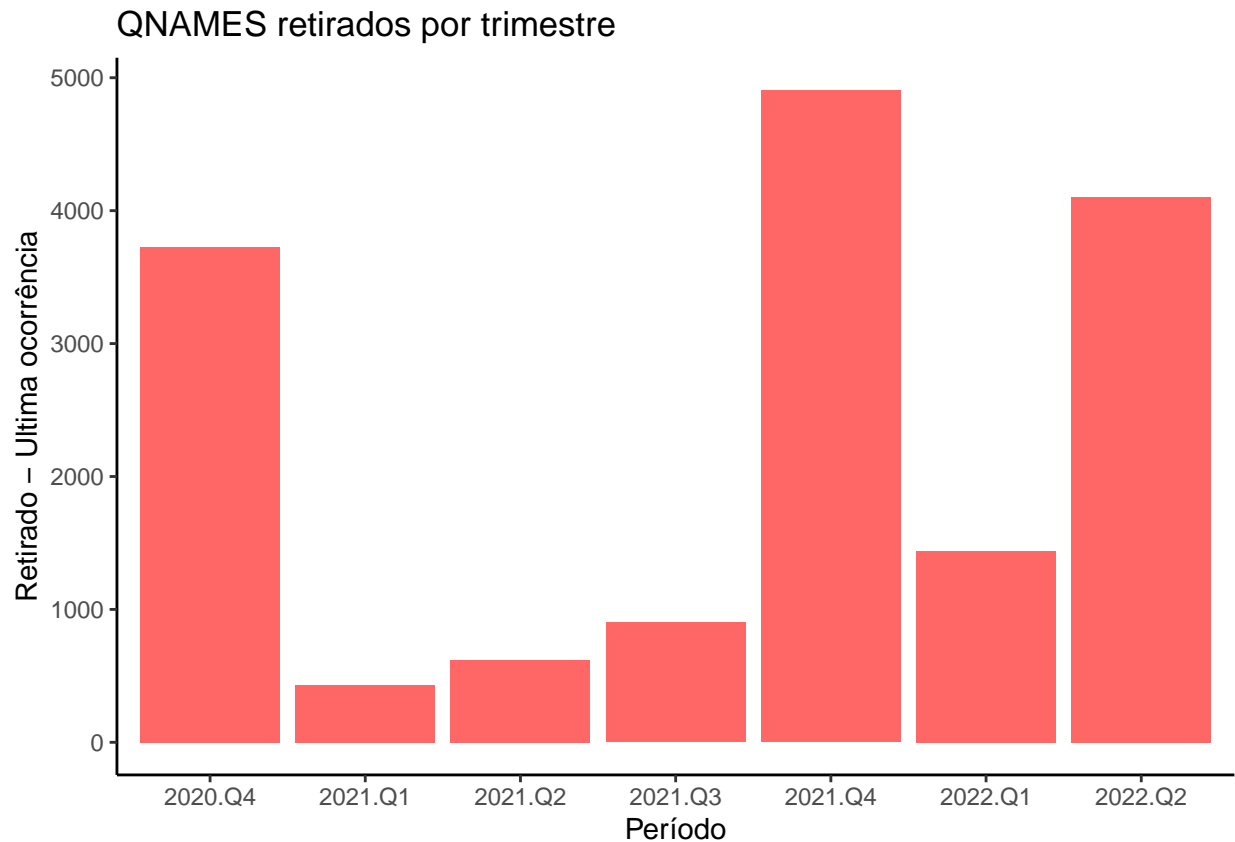
data_quarters_new_removed

```
## # A tibble: 7 x 3
##   year_period novos retirados
##   <fct>         <int>     <int>
## 1 2020.Q4         6637       3726
## 2 2021.Q1         1096        430
## 3 2021.Q2          306        618
## 4 2021.Q3          909        902
## 5 2021.Q4         4083       4905
## 6 2022.Q1          945       1440
## 7 2022.Q2         2149       4104
```

```
data_quarters_new_removed %>%
  ggplot( aes(x=year_period, y=novos)) +
  geom_bar(stat="identity", position="dodge", fill = "#00FF00") +
  ggtitle("Novos QNAMES por trimestre") +
  xlab("Período") +
  ylab("Novos - Primeira ocorrência") +
  theme_classic()
```



```
data_quarters_new_removed %>%
  ggplot( aes(x=year_period, y=retirados)) +
  geom_bar(stat="identity", position="dodge", fill = "#FF6666") +
  ggtitle("QNAMEs retirados por trimestre") +
  xlab("Período") +
  ylab("Retirado - Última ocorrência") +
  theme_classic()
```



QNAMEs Novos/Retirados por trimestre

- Um QNAME é considerado
 - Novo no trimestre em que aparece a primeira vez no database
 - Retirado no trimestre em que aparece pela ultima vez

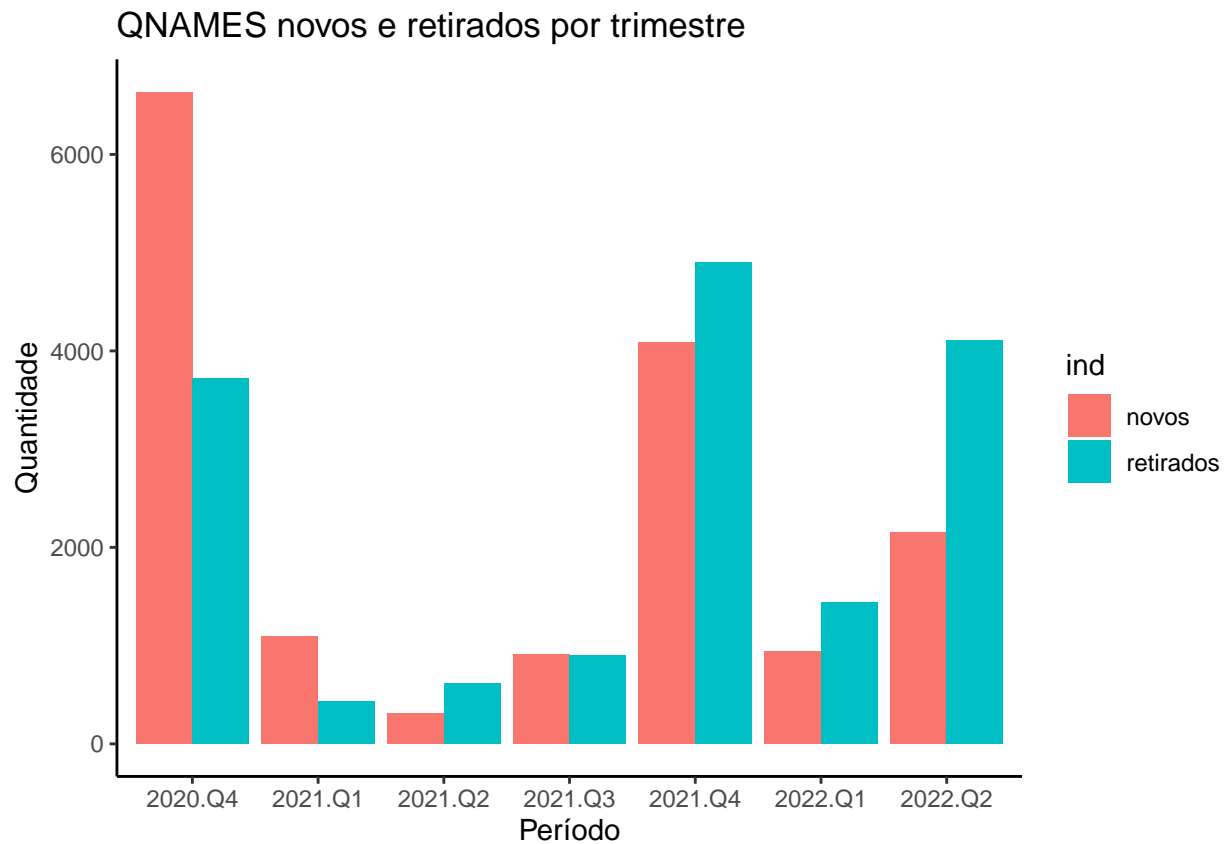
```
data_quarters_new_removed.st = stack(data_quarters_new_removed)
```

```
## Warning in stack.data.frame(data_quarters_new_removed): non-vector columns will
## be ignored
```

```
data_quarters_new_removed.st$year_period = rep(data_quarters_new_removed$year_period, 2)
```

```
data_quarters_new_removed.st %>%
  ggplot( aes(x=year_period, y=values, fill=ind)) +
```

```
geom_bar(stat="identity", position="dodge") +
ggtitle("QNAMES novos e retirados por trimestre") +
xlab("Período") +
ylab("Quantidade") +
theme_classic()
```



- Contudo, caso um ataque inicie e termine no mesmo trimestre, ele conta como um para novo e um para retirado
- Removendo todos os registros que iniciam e terminam no mesmo trimestre

```
data_different_quarters = data_quarters %>%
  filter(year_period_inicio != year_period_fim)

data_quarters_new2 = data_different_quarters %>%
  ungroup() %>%
  count(year_period_inicio, name = "novos")

data_quarters_removed2 = data_different_quarters %>%
  ungroup() %>%
  count(year_period_fim, name = "retirados")

data_diff_quarters_new_removed = data_quarters_new2 %>%
  inner_join(data_quarters_removed2, by = c("year_period_inicio" = "year_period_fim")) %>%
  mutate(year_period = as.factor(year_period_inicio)) %>%
  select('year_period', 'novos', 'retirados') %>%
```

```

arrange(year_period)

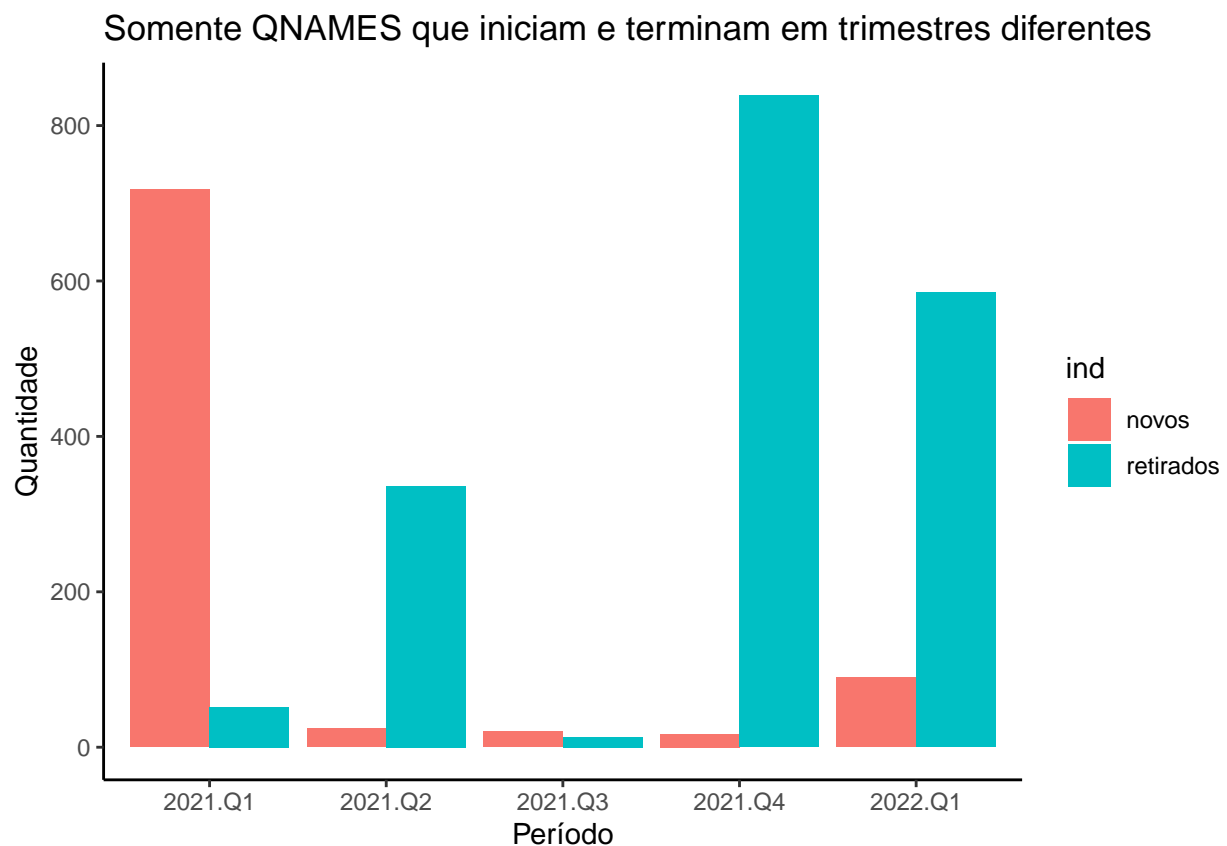
data_diff_quarters_new_removed.st = stack(data_diff_quarters_new_removed)

## Warning in stack.data.frame(data_diff_quarters_new_removed): non-vector columns
## will be ignored

data_diff_quarters_new_removed.st$year_period = rep(data_diff_quarters_new_removed$year_period, 2)

data_diff_quarters_new_removed.st %>%
  ggplot( aes(x=year_period, y=values, fill=ind)) +
  geom_bar(stat="identity", position="dodge") +
  ggtitle("Somente QNAMES que iniciam e terminam em trimestres diferentes") +
  xlab("Período") +
  ylab("Quantidade") +
  theme_classic()

```



- Dessa forma, foram considerados somente ataques em que iniciaram e finalizaram em trimestres diferentes

- Por curiosidade fui verificar para colocar lado a lado aqueles registros que iniciam e terminam no mesmo trimestre

```

data_same_quarters = data_quarters %>%
  filter(year_period_inicio == year_period_fim)

data_same_quarters = data_quarters %>%
  filter(year_period_inicio == year_period_fim) %>%
  ungroup() %>%
  count(year_period_inicio, name = "mesmo_trimestre") %>%
  mutate(year_period = as.factor(year_period_inicio) )

data_quarters_new_removed_same = data_diff_quarters_new_removed %>%
  inner_join(data_same_quarters, by = c("year_period" = "year_period")) %>%
  mutate(year_period = as.factor(year_period_inicio) ) %>%
  select('year_period', 'novos', 'retirados', 'mesmo_trimestre') %>%
  arrange(year_period)

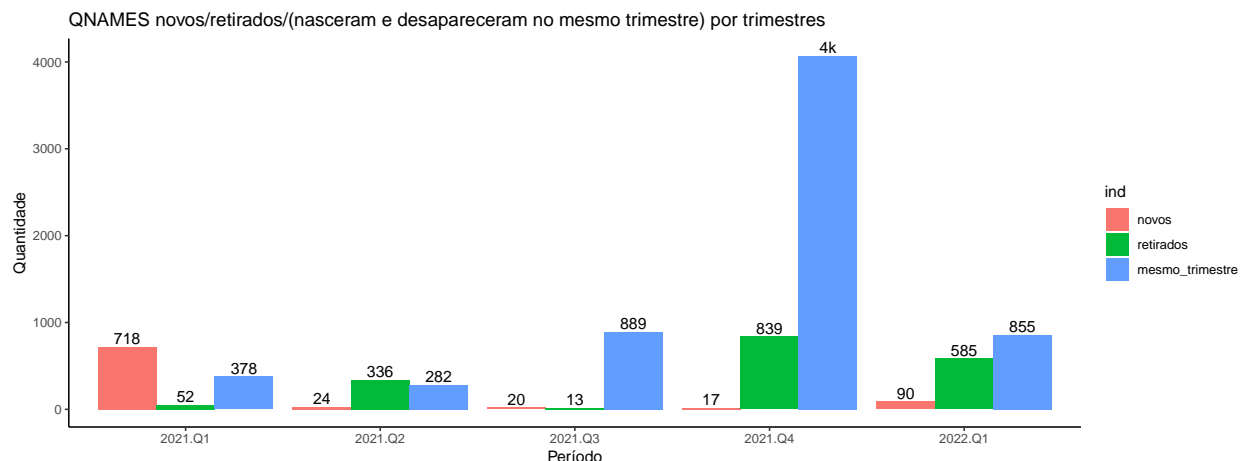
data_quarters_new_removed_same.st = stack(data_quarters_new_removed_same)

## Warning in stack.data.frame(data_quarters_new_removed_same): non-vector columns
## will be ignored

data_quarters_new_removed_same.st$year_period = rep(data_quarters_new_removed_same$year_period, 3)

data_quarters_new_removed_same.st %>%
  ggplot( aes(x=year_period, y=values, fill=ind)) +
  geom_bar(stat="identity", position=position_dodge()) +
  ggtitle("QNAMES novos/retirados/(nasceram e desapareceram no mesmo trimestre) por trimestres") +
  geom_text(aes(label = addUnits(values)), vjust=-0.25, position = position_dodge(0.9)) +
  # geom_text(aes(values, label = sprintf("%2.1f", values)), position = position_dodge(width = .9)) +
  # labs(color = "QNAME") +
  # scale_y_continuous(labels = addUnits) +
  xlab("Período") +
  ylab("Quantidade") +
  theme_classic()

```



- A barra azul então representa que no último trimestre de 2021, tiveram 4066 QNAMES que nasceram e morreram no mesmo período
- Média de requisições por ataque
 - A soma de todos os requests dos ataques agrupados por (QNAME+QTYPE) / dividido pela quantidade de ataques
 - Uma observação aqui é que muitos ataques que possuem uma média alta de requisições por ataque são aqueles em que tiveram apenas um ataque

```
data_quarters %>%
  ungroup() %>%
  arrange(desc(avg_requests_per_attack)) %>%
  select(avg_requests_per_attack, number_of_attacks, sum_requests_per_attack) %>%
  print(n=15)
```

```
## # A tibble: 16,125 x 3
##   avg_requests_per_attack number_of_attacks sum_requests_per_attack
##   <dbl>                <int>                <int>
## 1      18556.              25              463905
## 2      10499               2              20998
## 3       8467               1              8467
## 4       4953               1              4953
## 5       4537               1              4537
## 6       2794               1              2794
## 7       2222               1              2222
## 8       2026               1              2026
## 9       1830               1              1830
## 10      1702               1              1702
## 11      1544               4              6176
## 12      1485               1              1485
## 13      1462               1              1462
## 14      1395               1              1395
## 15      1221.              4              4883
## # ... with 16,110 more rows
```

```
quantile(data_quarters$avg_requests_per_attack, trim=trim_value)
```

```
##      0%      25%      50%      75%     100%
##    1.000    2.000    8.353   21.000 18556.200
```

- Média de ataques por dia
 - A média de ataques por dia, foi mensurado contando a quantidade de ataques (number_of_attacks) e dividindo pelo número de dias entre a data início e fim dos ataques de um (QNAME+QTYPE)

```
data_quarters_avg_attacks = data_quarters %>%
  ungroup() %>%
  mutate(tempo_diff_days=as.numeric(difftime(tempo_final, tempo_inicio, units = "days"))) %>%
  filter(tempo_diff_days > 1) %>%
  mutate(avg_attacks_per_day=number_of_attacks/tempo_diff_days)
```

```
data_quarters_avg_attacks %>%
  arrange(desc(avg_attacks_per_day)) %>%
  select(avg_attacks_per_day, avg_requests_per_attack, number_of_attacks, tempo_diff_days) %>%
  print(n=15)
```

```
## # A tibble: 4,109 x 4
##   avg_attacks_per_day avg_requests_per_attack number_of_attacks tempo_diff_days
##   <dbl>               <dbl>               <int>         <dbl>
## 1         5412.         16.5               49615         9.17
## 2          358.         29.0               56479        158.
## 3          256.         26.1               45523        178.
## 4          213.        615.              118333        557.
## 5          112.         45.7               58009        520.
## 6           55.8        197.               30375        544.
## 7           42.8         1.62                74         1.73
## 8           42.0         2.74                42         1.00
## 9           41.4         1.67                143         3.46
## 10          39.4         2.00                64         1.62
## 11          38.7         1.39                67         1.73
## 12          38.7         1.91                67         1.73
## 13          38.0         3.45                38         1.00
## 14          37.7         1.96                193         5.12
## 15          37.6         1.8                 65         1.73
## # ... with 4,094 more rows
```

```
data_quarters_avg_attacks.count = data_quarters_avg_attacks %>%
  count(avg_attacks_per_day, name = "number_of_qnames") %>%
  arrange(desc(number_of_qnames))

data_quarters_avg_attacks.count %>%
  print(n=10)
```

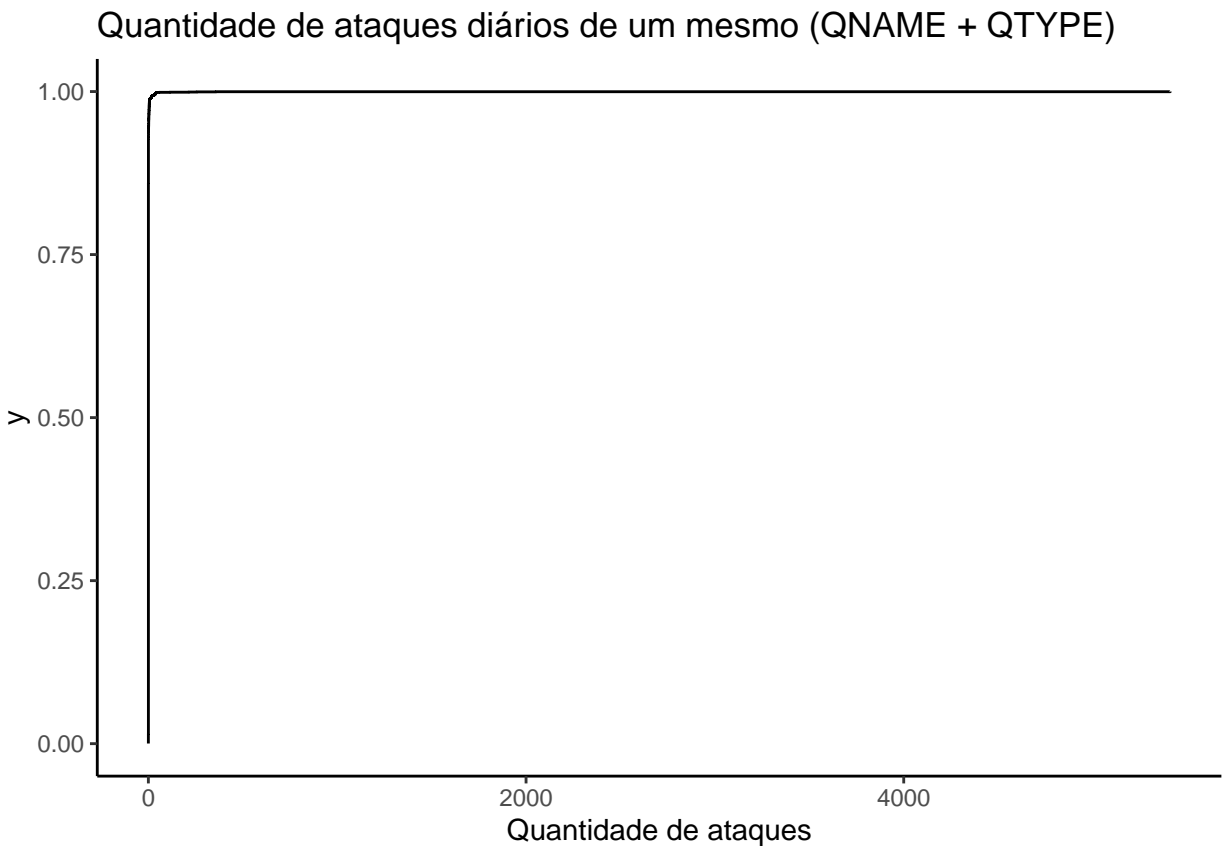
```
## # A tibble: 4,109 x 2
##   avg_attacks_per_day number_of_qnames
##   <dbl>               <int>
## 1         0.00582         1
## 2         0.00676         1
## 3         0.00680         1
## 4         0.00719         1
## 5         0.00735         1
## 6         0.00749         1
## 7         0.00752         1
## 8         0.00767         1
## 9         0.00808         1
## 10        0.00876         1
## # ... with 4,099 more rows
```

```
quantile(
  data_quarters_avg_attacks$avg_attacks_per_day[data_quarters_avg_attacks$avg_attacks_per_day>0],
  trim=trim_value
)
```

##	0%	25%	50%	75%	100%
##	0.005822	0.100274	0.115973	0.139103	5411.568366

- Em 3.868 QNAMES a média de ataques por dia é 0.1391, 63.65 % dos registros
- mas em 25% dos QNAMES possuem uma média de mais de 17 ataques por dia

```
data_quarters_avg_attacks %>%
  ggplot(aes(x= avg_attacks_per_day)) +
  stat_ecdf(geom = "step", pad = FALSE) +
  ggtitle("Quantidade de ataques diários de um mesmo (QNAME + QTYPE)") +
  xlab("Quantidade de ataques") +
  theme_classic()
```



- Obelheiro
- A propósito, acho que pode ser interessante verificar:
 - quantos nomes são usados em apenas um ataque;
 - quantos nomes aparecem e desaparecem no mesmo dia, ou dentro de 24h (“mesmo dia” pode ser influenciado pelo timezone)

```
data %>%
  count(qname) %>%
  filter(n == 1) %>%
  count(n) %>%
  select(nn)
```

```
## Storing counts in 'nn', as 'n' already present in input
## i Use 'name = "new_name"' to pick a new name.
```

```
##      nn
## 1 11636
```

- 11.636 qnames foram utilizados em apenas 1 ataque

```
dns_data_longevidade = data %>%
  group_by(qname) %>%
  summarise(tempo_inicio=min(tempo_inicio_cast), tempo_final=max(tempo_final_cast),
            number_of_attacks=n(), year_period = first(year_period)) %>%
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"),
         tempo_diff = tempo_final - tempo_inicio)

dns_data_longevidade %>%
  filter(tempo_diff_secs < days) %>%
  group_by(qname) %>%
  summarise(qname_count = n()) %>%
  group_by(qname_count) %>%
  summarise(qname_qnt_attack_24h = n())
```

```
## # A tibble: 1 x 2
##   qname_count qname_qnt_attack_24h
##       <int>          <int>
## 1           1             11924
```

- 11.924 qnames tiveram ataques que duram somente 1 dia
- Quantos qnames existem = 15.305

```
data %>%
  group_by(qname) %>%
  summarise(qname_count = n())
```

```
## # A tibble: 15,305 x 2
##   qname                                qname_count
##   <chr>                                <int>
## 1 _sip._tls.sinastorage.com.             1
## 2 .                                9850
## 3 0_10_3356716014.ninini.xyz.             1
## 4 0_3356716014.ninini.xyz.                 3
## 5 0_4_3356716014.ninini.xyz.                 1
## 6 0.200-19-107-238.dualstack.xiaofengtest.com. 2
## 7 0.africa.pool.ntp.org.                   2
## 8 0.asia.pool.ntp.org.                       7
## 9 0.au.pool.ntp.org.                         5
## 10 0.cn.pool.ntp.org.                       3
## # ... with 15,295 more rows
```

```

dns_qnames = inner_join(
  by = "year_period",
  dns_data_longevidade %>%
    ungroup() %>%
    group_by(year_period) %>%
    summarise("total_qnames" = n())
,
  dns_data_longevidade %>%
    ungroup() %>%
    filter(tempo_diff_secs < days) %>%
    group_by(year_period) %>%
    summarise("qname_one_day" = n())
)

stack_qnames = stack(dns_qnames %>%
  mutate(year_period = as.factor(year_period)))

```

```

## Warning in stack.data.frame(dns_qnames %>% mutate(year_period =
## as.factor(year_period))): non-vector columns will be ignored

```

```
stack_qnames$year_period = rep(dns_qnames$year_period, 2)
```

- Trimestres e rate de ataques que duram um dia pelo total de qnames

```

dns_qnames %>%
  mutate(rate_one_day = (qname_one_day / total_qnames) * 100)

```

```

## # A tibble: 7 x 4
##   year_period total_qnames qname_one_day rate_one_day
##   <int>         <int>         <int>         <dbl>
## 1     20204         6034         3520         58.3
## 2     20211          908          323         35.6
## 3     20212          303          254         83.8
## 4     20213          904          884         97.8
## 5     20214         4075         4049         99.4
## 6     20221          942          826         87.7
## 7     20222         2139         2068         96.7

```

- Calculando totais

```

dns_qnames_one_day_all = dns_qnames %>%
  mutate(sum_total_qnames = sum(total_qnames), sum_qname_one_day = sum(qname_one_day)) %>%
  select('sum_total_qnames', 'sum_qname_one_day') %>%
  head(1) %>%
  mutate(rate_one_day = (sum_qname_one_day / sum_total_qnames) * 100)

dns_qnames_one_day_all %>%
  print(n=1)

```

```

## # A tibble: 1 x 3
##   sum_total_qnames sum_qname_one_day rate_one_day
##   <int>         <int>         <dbl>
## 1     15305         11924         77.9

```

- 77.9092% dos qnames possuem um ataque de duração de um dia
- 15.305 qnames aparecem e desaparecem no mesmo dia, ou dentro de 24h (“mesmo dia”)

```
stack_qnames %>%
  mutate(year_period = paste(substr(year_period, 0, 4), substr(year_period, 5, 5), sep = "/")) %>%
  ggplot(aes(x=year_period, y=values, fill=ind)) +
  geom_bar(stat="identity", position="dodge") +
  # geom_bar(stat="identity", position="dodge", fill = "#FF6666") +
  # geom_text(aes(label = values), vjust = 2, yjust = 4) +
  geom_text(aes(label = values), vjust=-0.25, position = position_dodge(0.9)) +
  scale_y_continuous(labels = addUnits) +
  ggtitle("QNAMES aparecem e desaparecem no mesmo dia por trimestre") +
  xlab("Trimestre") +
  ylab("qnames que aparecem e desaparecem no mesmo dia") +
  theme_classic()
```

