

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 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="../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_month = (60 * 60 * 24 * 30)
```

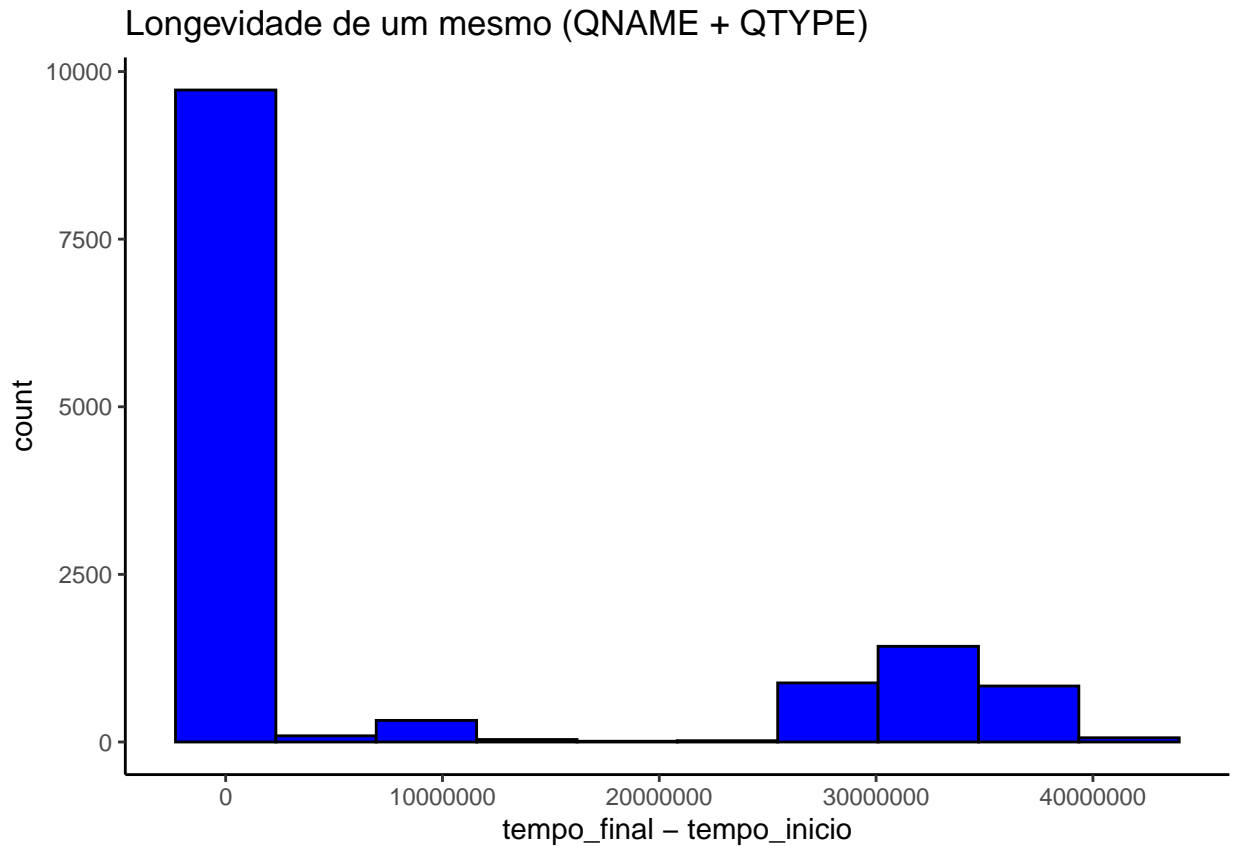
```
data_grouped = data %>%
  group_by(qname, qtype) %>%
  summarise(tempo_inicio=min(tempo_inicio_cast), tempo_final=max(tempo_final_cast), sum_requests_per_at~
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_fin~
  #filter(tempo_diff > 0) %>%
  arrange(desc(tempo_diff_secs))
```

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

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

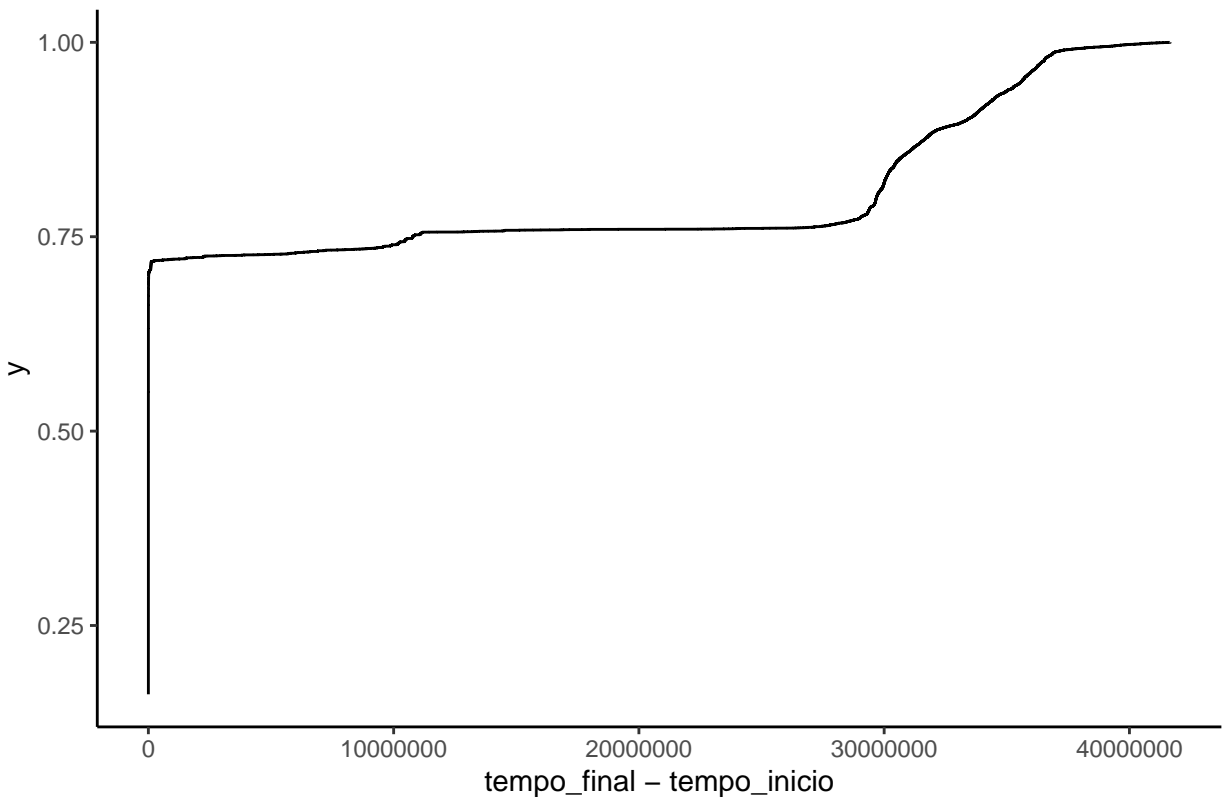
```
## # A tibble: 10 x 7
## # Groups:   qname [10]
##   qname          qtype tempo_inicio      tempo_final      sum_requests_pe~
##   <chr>         <chr> <dtm>          <dtm>          <int>
## 1 VERSION.BIND. TXT   2020-10-30 02:39:27 2022-02-24 08:33:01      5943
## 2 whoami.akamai~ A     2020-10-30 12:33:16 2022-02-23 10:35:19      3885
## 3 com.          ANY   2020-10-31 11:38:32 2022-02-24 09:16:50       478
## 4 version.bind. TXT   2020-11-01 04:46:10 2022-02-24 15:42:17       592
## 5 isc.org.      ANY   2020-11-01 22:40:23 2022-02-24 02:25:52    2652144
## 6 researchscan5~ A     2020-11-01 15:23:29 2022-02-23 14:06:13       147
## 7 adsports.ae.  MX    2020-11-03 10:59:02 2022-02-23 23:36:50     2452
## 8 public1.114dn~ A     2020-11-04 09:47:54 2022-02-24 15:42:14     1764
## 9 238.107.19.20~ PTR   2020-11-01 23:58:38 2022-02-21 13:45:36       627
## 10 a.gtld-server~ A     2020-10-30 02:55:07 2022-02-18 00:20:55        78
## # ... with 2 more variables: tempo_diff_secs <dbl>, tempo_diff <drtn>
```

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



```
data_grouped %>%
  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()
```

Longevidade de um mesmo (QNAME + QTYPE)



```
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      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      sum_requests_per_attack tempo_diff_secs
## Min.   :2020-10-29 23:17:13      Min.   :      1      Min.   :      0
## 1st Qu.:2020-12-15 00:54:54      1st Qu.:      3      1st Qu.:     14
## Median :2021-10-25 06:53:39      Median :     12      Median :     69
## Mean   :2021-07-31 07:23:43      Mean   :    6638      Mean   : 8167044
```

```
## 3rd Qu.:2021-12-06 22:50:36 3rd Qu.: 161 3rd Qu.:10780940
## Max. :2022-02-24 15:49:24 Max. :72346023 Max. :41666014
## tempo_diff
## Length:13411
## Class :difftime
## Mode :numeric
##
##
##
```

- Dados sobre o intervalo entre a primeira e a última aparição desse (QNAME+QTYPE)
 - Mínimo 0 segundos
 - Máximo 16.0749 meses
 - Média 136117.3974 minutos
 - Mediana 69 segundos

```
trim_value = .30

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       14       69 10780940 41666014
```

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

```
##      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      sum_requests_per_attack tempo_diff_secs
## Min. :2020-10-29 23:17:13      Min. : 1      Min. : 0
## 1st Qu.:2020-12-15 00:54:54      1st Qu.: 3      1st Qu.: 14
## Median :2021-10-25 06:53:39      Median : 12     Median : 69
## Mean :2021-07-31 07:23:43      Mean : 6638     Mean : 8167044
## 3rd Qu.:2021-12-06 22:50:36      3rd Qu.: 161     3rd Qu.:10780940
## Max. :2022-02-24 15:49:24      Max. :72346023     Max. :41666014
##      tempo_diff
## Length:13411
## Class :difftime
## Mode :numeric
##
##
##
```

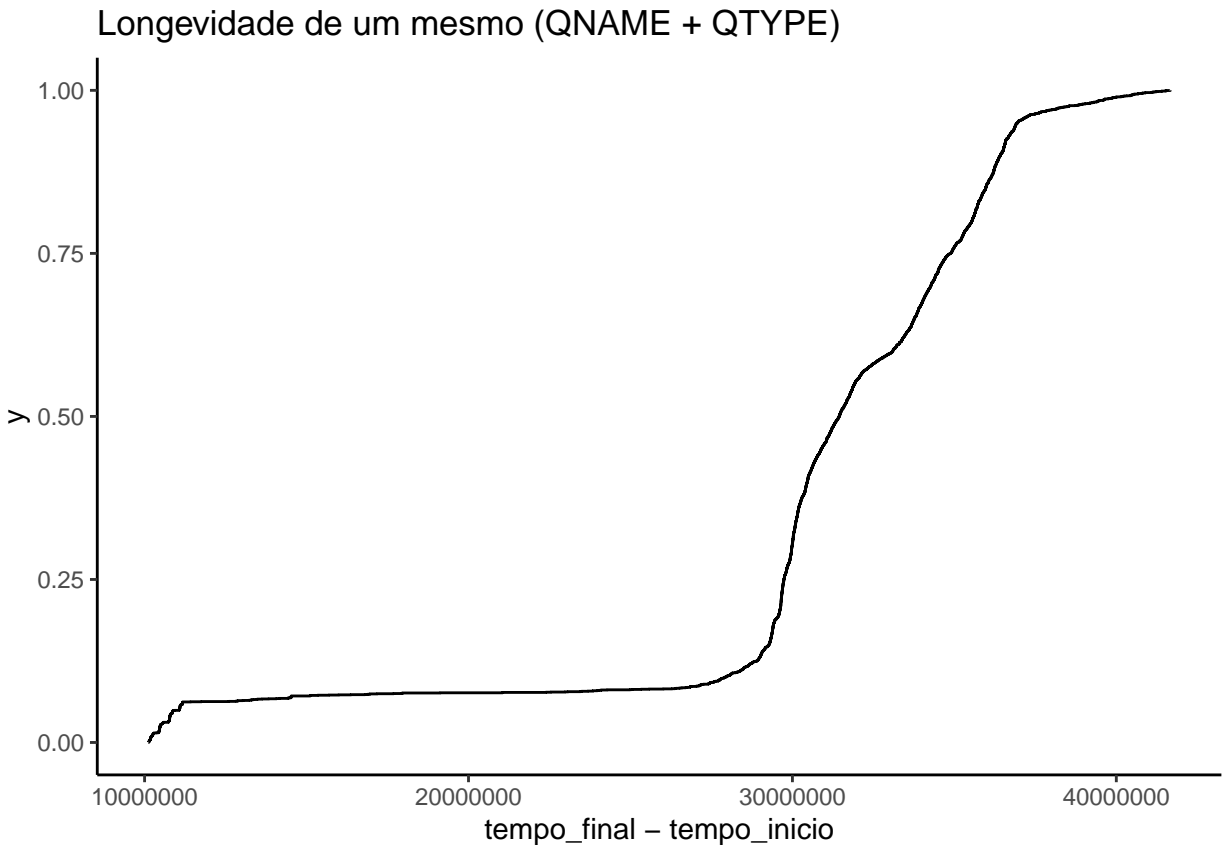
- Dados sobre o intervalo entre a primeira e a última aparição do (QNAME+QTYPE) removendo 30% dos valores máximos e mínimos
 - Mínimo 0 segundos
 - Máximo 16.0749 meses
 - Média 12.7382 minutos
 - Mediana 69 segundos
- Identificar qual é o percentil em que há essa mudança de tendência (próximo aos 72-73%) 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% 69.63% 71.03% 72% 72.04% 72.11% 73%
## 100 1000 10000 100000 598289 803076 1000000 6310057
## 73.97% 75% 76% 77% 78% 99%
## 10000000 10780940 23306177 28575350 29310516 37308538
```

- 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 cresce muito até cerca de 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)
 - Possivelmente apresenta uma distribuição assimétrica com cauda a direita

```
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_month
```

```
## [1] 8.992
```

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

```
## [1] 15.12
```

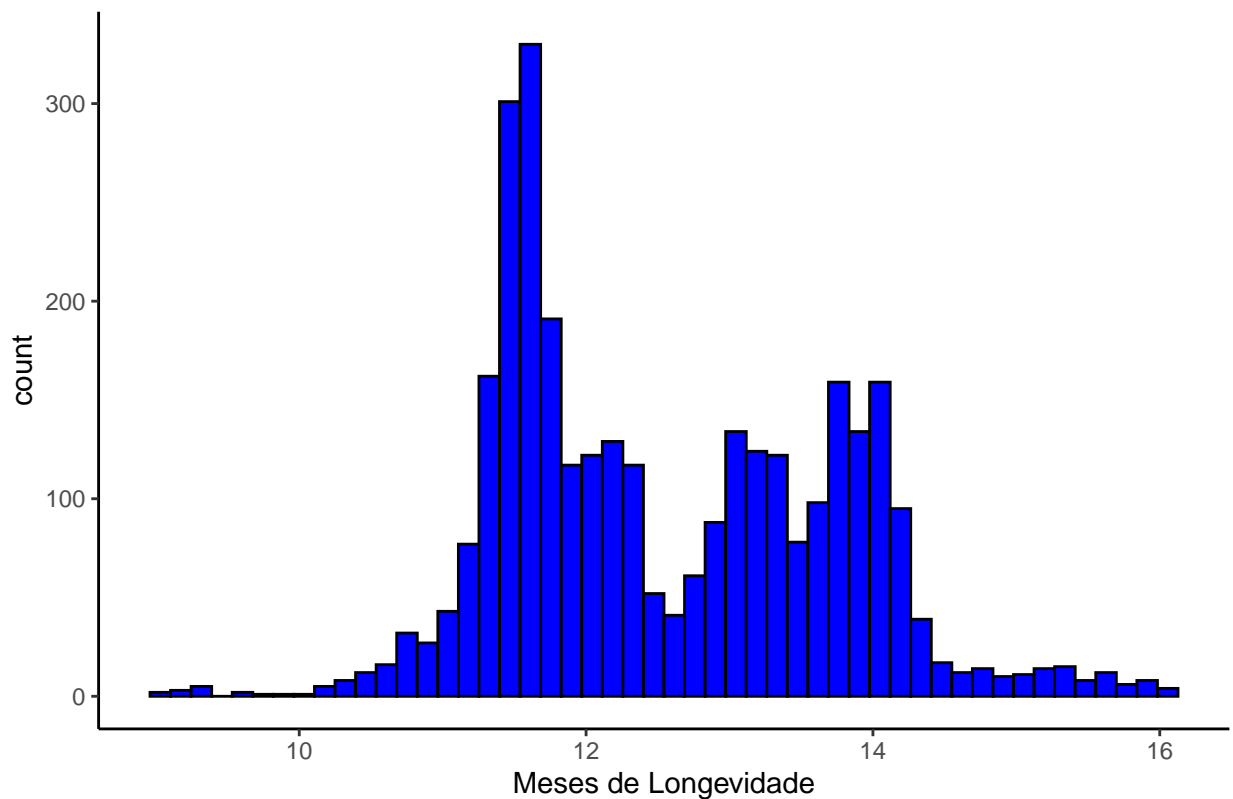
```
data_grouped$tempo_diff_secs.max / secs_to_month
```

```
## [1] 16.07
```

- 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_month)) +
  geom_histogram(bins = 50, fill='blue', color='black') +
  ggtitle("Longevidade de um mesmo (QNAME + QTYPE)") +
  xlab("Meses 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))

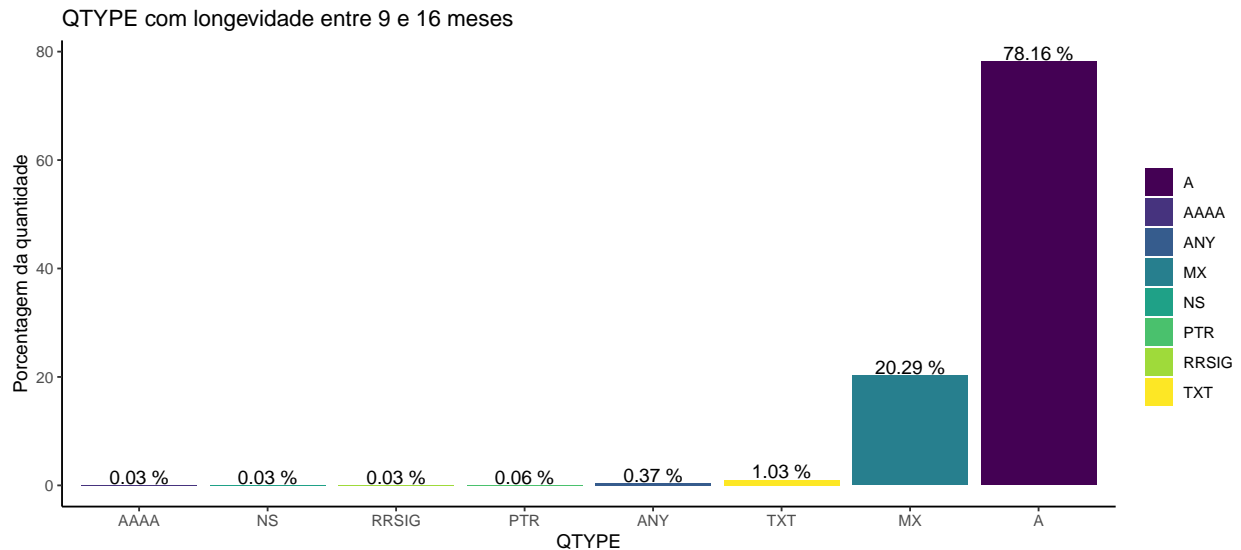
sum_qtype_quantity = sum(data_bigger_than_76$qtype_quantity)
data_bigger_than_76_percentage = data_bigger_than_76 %>%
  mutate(qtype_quantity_percentage = (qtype_quantity / sum_qtype_quantity) * 100)

data_bigger_than_76_percentage
```

```
## # A tibble: 8 x 3
##   qtype qtype_quantity qtype_quantity_percentage
##   <chr>         <int>             <dbl>
## 1 A             2516             78.2
## 2 MX             653             20.3
## 3 TXT             33              1.03
## 4 ANY             12              0.373
## 5 PTR              2              0.0621
## 6 AAAA             1              0.0311
## 7 NS               1              0.0311
## 8 RRSIG            1              0.0311
```



```
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% (653) deles possuem o QTYPE “MX”
 - 78% (2516) dos ataques com maior longevidade utilizam o QTYPE “A”, o que é surpreendente
 - E por fim o QTYPE “ANY” aparece com apenas 12 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),
  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(tempo_diff_secs)) %>%
  head(10)
```

```
## # A tibble: 10 x 7
##   qname      qname_count sum_requests_per_attack tempo_inicio      tempo_final
##   <chr>          <int>          <int> <dtm>          <dtm>
## 1 whoami.akamai.net 3639          3885 2020-10-30 12:33:16 2022-02-23 10:35:19
## 2 research.att.com 143           147 2020-11-01 15:23:29 2022-02-23 14:06:13
## 3 public1.safelink.net 479          1764 2020-11-04 09:47:54 2022-02-24 15:42:14
## 4 a.gtld.safelink.net 73           78 2020-10-30 02:55:07 2022-02-18 00:20:55
## 5 dnsscan.safelink.net 67           71 2020-11-05 15:30:18 2022-02-22 00:35:12
## 6 amazon.com 433          2014 2020-11-10 15:16:10 2022-02-24 06:20:17
## 7 www.bb.com 24           51 2020-11-05 15:04:15 2022-02-17 15:57:08
## 8 www.baidu.com 17           28 2020-11-11 03:58:33 2022-02-19 11:34:15
## 9 www.google.com 13           40 2020-11-19 09:44:27 2022-02-23 13:34:35
## 10 www.brazil.br 5            5 2020-11-13 22:09:40 2022-02-17 16:27:09
## # ... with 2 more variables: tempo_diff_secs <dbl>, tempo_diff <drtn>
```

- Única 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)) %>%
  head(10)
```

```
## # A tibble: 10 x 7
##   qname      qname_count sum_requests_per_attack tempo_inicio      tempo_final
##   <chr>          <int>          <int> <dtm>          <dtm>
## 1 admin.a.com 50          21565 2020-12-28 05:54:53 2022-01-29 06:11:28
## 2 theguardian.com 58          12731 2021-02-07 10:19:59 2022-02-24 11:15:23
## 3 ftp.ebi.ac.uk 40          11840 2020-12-25 00:28:07 2022-02-18 08:57:08
## 4 dji.com 57          11246 2020-12-27 01:14:12 2022-02-23 13:30:12
## 5 hotspot.com 44          11004 2021-02-07 10:19:50 2021-12-31 16:37:20
## 6 emarata.com 50          9447 2020-12-20 06:29:54 2021-12-12 02:55:22
## 7 vpn.qat.net 44          6651 2021-01-04 10:52:05 2022-02-09 23:24:32
## 8 moi.gov.et 43          6280 2021-01-01 14:29:59 2022-01-16 07:14:32
## 9 tmall.com 13          5625 2020-11-25 10:50:33 2022-01-28 09:18:56
## 10 vr1.mynl.com 50          4896 2020-12-28 14:22:21 2022-02-08 23:01:33
## # ... with 2 more variables: tempo_diff_secs <dbl>, tempo_diff <drtn>
```

- 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_final),
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_diff_secs),
  arrange(desc(query_id_count)) %>%
  head(10)
```

```
## # A tibble: 10 x 7
##   query_id query_id_count sum_requests_per_attack tempo_inicio
##   <int>      <int>          <int> <dtm>
## 1    50265          618            1283 2020-12-18 04:26:58
## 2    28826          145             167 2021-03-01 10:34:11
## 3     4218           98             110 2020-10-30 18:46:37
## 4     1337           74              79 2020-10-30 02:55:07
## 5    44557           61              68 2020-11-26 11:07:27
## 6    64206           57              59 2021-02-05 13:23:20
## 7       256           43             417 2021-01-19 02:50:02
## 8    16028           40            3049 2020-11-10 02:29:52
## 9         0           31             216 2020-11-03 07:26:10
## 10   19205           18              74 2020-11-19 09:44:27
## # ... with 3 more variables: tempo_final <dtm>, tempo_diff_secs <dbl>,
## #   tempo_diff <drtn>
```

- Nada chamou a atenção

```
#N = 10
```

```
#data_split_year_period = data %>%
#   group_split(year_period)

#period_query_id_qname = data.frame()
#for (i in c(1:length(data_split_year_period))) {
#   query_id_qname_frequency = data_split_year_period[[i]] %>%
#     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))
#     mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_diff_secs),
#     arrange(desc(tempo_diff_secs))
#   }
# period_query_id_qname = rbind(period_query_id_qname, head(query_id_qname_frequency, N))
#}
```

```
data %>%
  group_by(year_period, qtype) %>%
  summarise(sum_grouped_year_period_qtype = n()) %>%
  group_by(year_period) %>%
  mutate(sum_qtype_year_period = sum(sum_grouped_year_period_qtype), qtype_percentage = ((sum_grouped_year_period_qtype / sum_grouped_year_period_qtype) * 100))
```

```
arrange(desc(qtype_percentage)) %>%
head(10)
```

Registros separados por trimestre

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

```
## # A tibble: 10 x 5
## # Groups:   year_period [6]
##   year_period qtype sum_grouped_year_period_~ sum_qtype_year_~ qtype_percentage
##         <int> <chr>                <int>          <int>          <dbl>
## 1      20204 ANY                122840        136983          89.7
## 2      20211 ANY                162599        213052          76.3
## 3      20221 ANY                 6934         11595          59.8
## 4      20212 ANY                23182         39611          58.5
## 5      20213 A                  30519         53934          56.6
## 6      20214 A                  47906         98317          48.7
## 7      20214 ANY                39673         98317          40.4
## 8      20221 A                   3826         11595          33.0
## 9      20212 A                  12326         39611          31.1
## 10     20211 A                  38499        213052          18.1
```

- Agrupa os dados por (qname, qtype, year_period) para calcular por trimestre a longevidade dos registros com o mesmo QNAME+QTYPE
 - OBS.: Inicialmente eu iria verificar a longevidade, mas não faz sentido então separar por trimestre para verificar o quanto um qname se manteve ativo dentro do trimestre, correto? @Obelheiro @Thiago

```
data_year_period = data %>%
  group_by(qname, qtype, year_period) %>%
  summarise(tempo_inicio=min(tempo_inicio_cast), tempo_final=max(tempo_final_cast), sum_requests_per_at=
  mutate(year_period=as.factor(year_period), tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, u
  arrange(desc(tempo_diff_secs))
```

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

- Agrupa somente pelo trimestre e o QTYPE para calcular quantos qnames diferentes existem em cada QTYPE

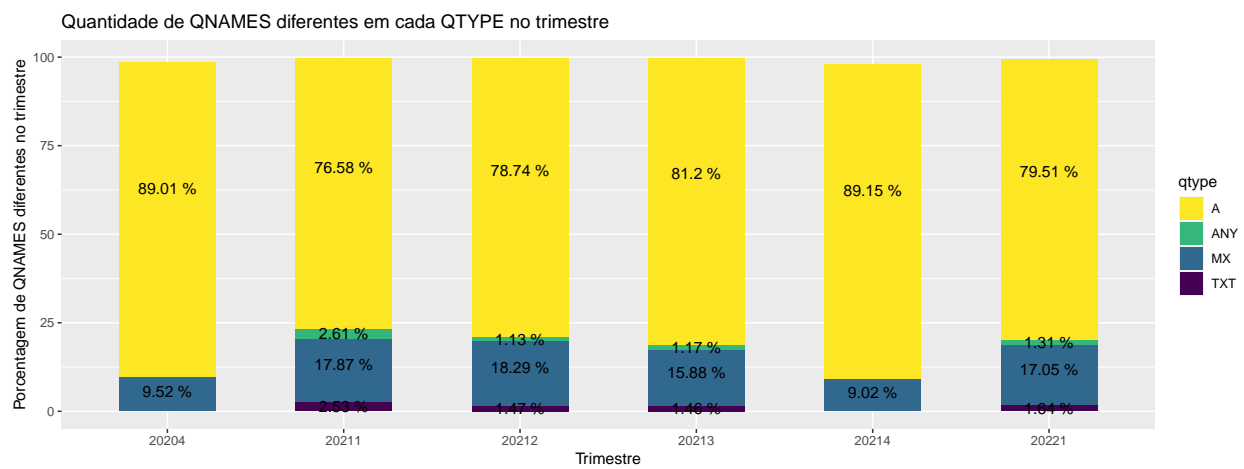
```
data_year_period_qtype_quantity = data_year_period %>%
  ungroup() %>%
  group_by(year_period, qtype) %>%
  summarise(qtype_quantity = n())
```

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

- Calcular a porcentagem da quantidade QTYPE

```
data_year_period_qtype_quantity_percentage = data_year_period_qtype_quantity %>%
  group_by(year_period) %>%
  mutate(sum_qtype_quantity_year_period = sum(qtype_quantity), qtype_year_period_quantity_percentage=((
```

```
data_year_period_qtype_quantity_percentage %>%
  filter(qtype_year_period_quantity_percentage > 1) %>%
  ggplot( aes(x=year_period, y=qtype_year_period_quantity_percentage, fill=qtype)) +
  geom_bar(stat="identity", width = 0.55) +
  geom_text(aes(label = paste(round(qtype_year_period_quantity_percentage, 2), "%")), position = posi
  scale_fill_viridis(discrete=TRUE, direction = -1) +
  ylab("Porcentagem de QNAMES diferentes no trimestre") +
  xlab("Trimestre") +
  ggtitle("Quantidade de QNAMES diferentes em cada QTYPE no trimestre")
```



- Isso mostra que o QTYPE “A” tem uma grande quantidade de ataques com QNAMES distintos em cada trimestre