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="../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_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_fina
  arrange(desc(tempo_diff_secs))
```

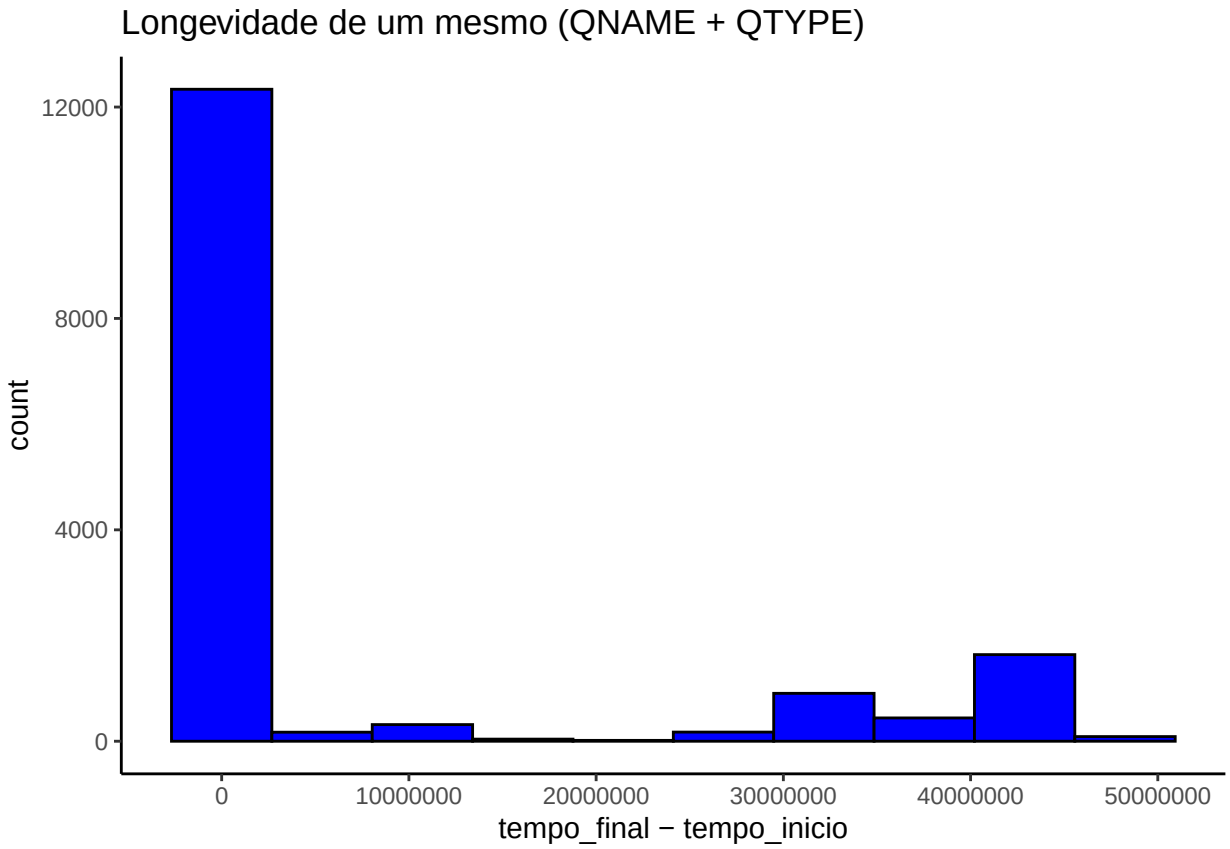
```
## `summarise()` has grouped output by 'qname'. You can override using the
## `.groups` argument.
```

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

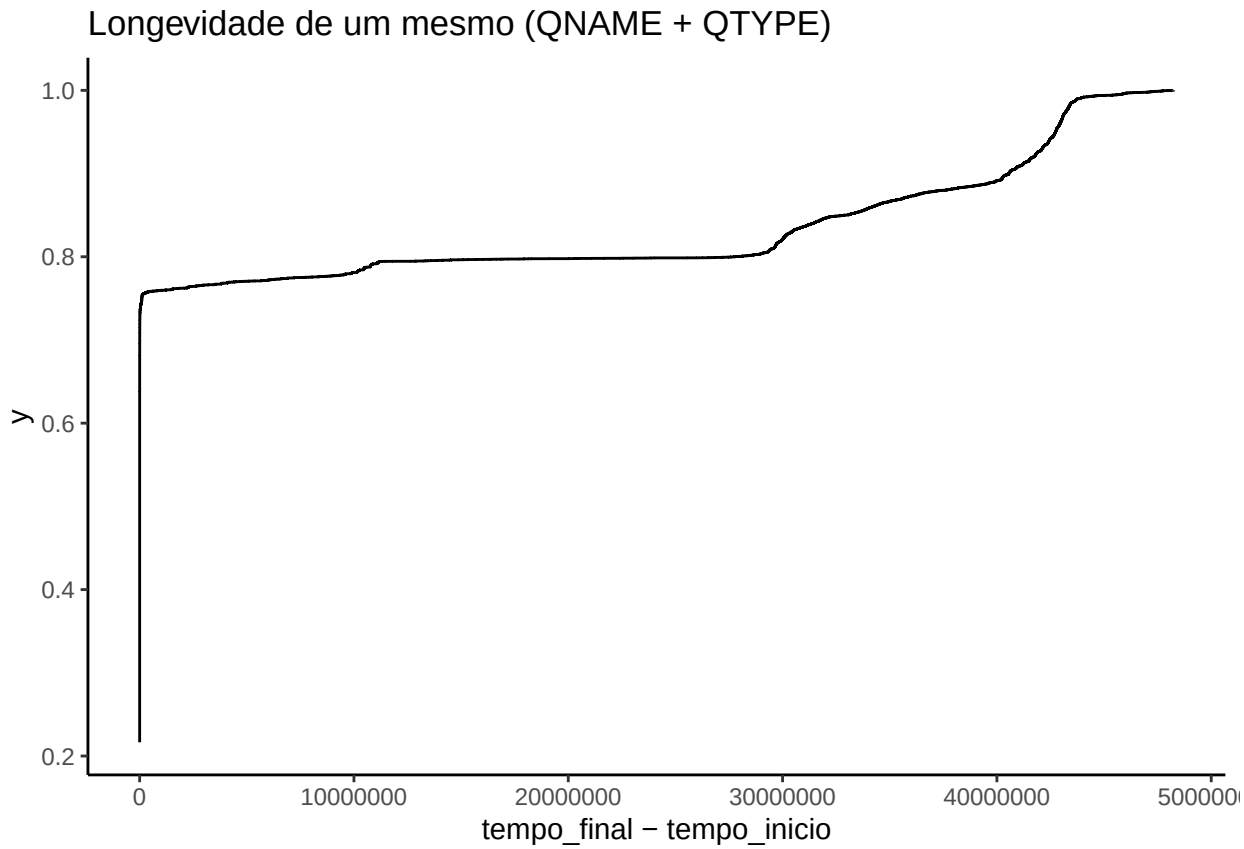
```
## # A tibble: 10 x 7
## # Groups:   qname [10]
##    qname          qtype tempo_inicio        tempo_final         sum_requests_pe~
##    <chr>          <chr> <dttm>              <dttm>                         <int>
##  1 VERSION.BIND.  TXT   2020-10-30 02:39:27 2022-05-11 13:05:22            6160
##  2 peacecorps.go~ ANY   2020-10-31 14:28:23 2022-05-11 05:22:49        72734156
##  3 200-19-107-23~ A     2020-10-30 11:15:36 2022-05-09 21:23:22             148
##  4 version.bind.  TXT   2020-11-01 04:46:10 2022-05-11 12:13:52             833
##  5 a.gtld-server~ A     2020-10-30 02:55:07 2022-05-09 06:50:12              89
##  6 com.           ANY   2020-10-31 11:38:32 2022-05-09 16:38:23             511
##  7 238.107.19.20~ PTR   2020-11-01 23:58:38 2022-05-09 04:56:13             842
##  8 dns-test.rese~ TXT   2020-11-04 06:49:48 2022-05-11 09:58:47              87
##  9 mopa.ae.       MX    2020-11-04 14:39:23 2022-05-11 13:52:05            6096
## 10 szgmc.gov.ae.  MX    2020-11-03 14:47:10 2022-05-10 13:29:54             889
## # ... 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()
```




```
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
##  Class :character   Class :character   1st Qu.:2020-12-17 16:50:24
##  Mode  :character   Mode  :character   Median :2021-07-09 00:38:37
##                                        Mean   :2021-06-25 05:28:30
##                                        3rd Qu.:2021-11-15 13:56:59
##                                        Max.   :2022-05-11 13:22:23
##   tempo_final                  sum_requests_per_attack tempo_diff_secs
##  Min.   :2020-10-29 23:17:13   Min.   :       1        Min.   :       0
##  1st Qu.:2021-03-08 14:18:01   1st Qu.:       2        1st Qu.:       5
##  Median :2021-11-01 10:55:05   Median :       9        Median :      67
##  Mean   :2021-09-25 18:50:31   Mean   :    5580        Mean   : 7996921
```

```
##  3rd Qu.:2022-04-02 11:48:10   3rd Qu.:     101        3rd Qu.:  105178
##  Max.   :2022-05-11 14:41:27   Max.   :72734156        Max.   :48248755
##   tempo_diff
##  Length:16125
##  Class :difftime
##  Mode  :numeric
##
##
##
```

- Dados sobre o intervalo entre a primeira e a última aparição desse (QNAME+QTYPE)
  - Mínimo 0 segundos
  - Máximo 18.6145 meses
  - Média 133282.019 minutos
  - Mediana 67 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        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
##  Class :character   Class :character   1st Qu.:2020-12-17 16:50:24
##  Mode  :character   Mode  :character   Median :2021-07-09 00:38:37
##                                        Mean   :2021-06-25 05:28:30
##                                        3rd Qu.:2021-11-15 13:56:59
##                                        Max.   :2022-05-11 13:22:23
##   tempo_final                  sum_requests_per_attack tempo_diff_secs
##  Min.   :2020-10-29 23:17:13   Min.   :       1        Min.   :       0
##  1st Qu.:2021-03-08 14:18:01   1st Qu.:       2        1st Qu.:       5
##  Median :2021-11-01 10:55:05   Median :       9        Median :      67
##  Mean   :2021-09-25 18:50:31   Mean   :    5580        Mean   : 7996921
##  3rd Qu.:2022-04-02 11:48:10   3rd Qu.:     101        3rd Qu.:  105178
##  Max.   :2022-05-11 14:41:27   Max.   :72734156        Max.   :48248755
##   tempo_diff
##  Length:16125
##  Class :difftime
##  Mode  :numeric
##
##
##
```

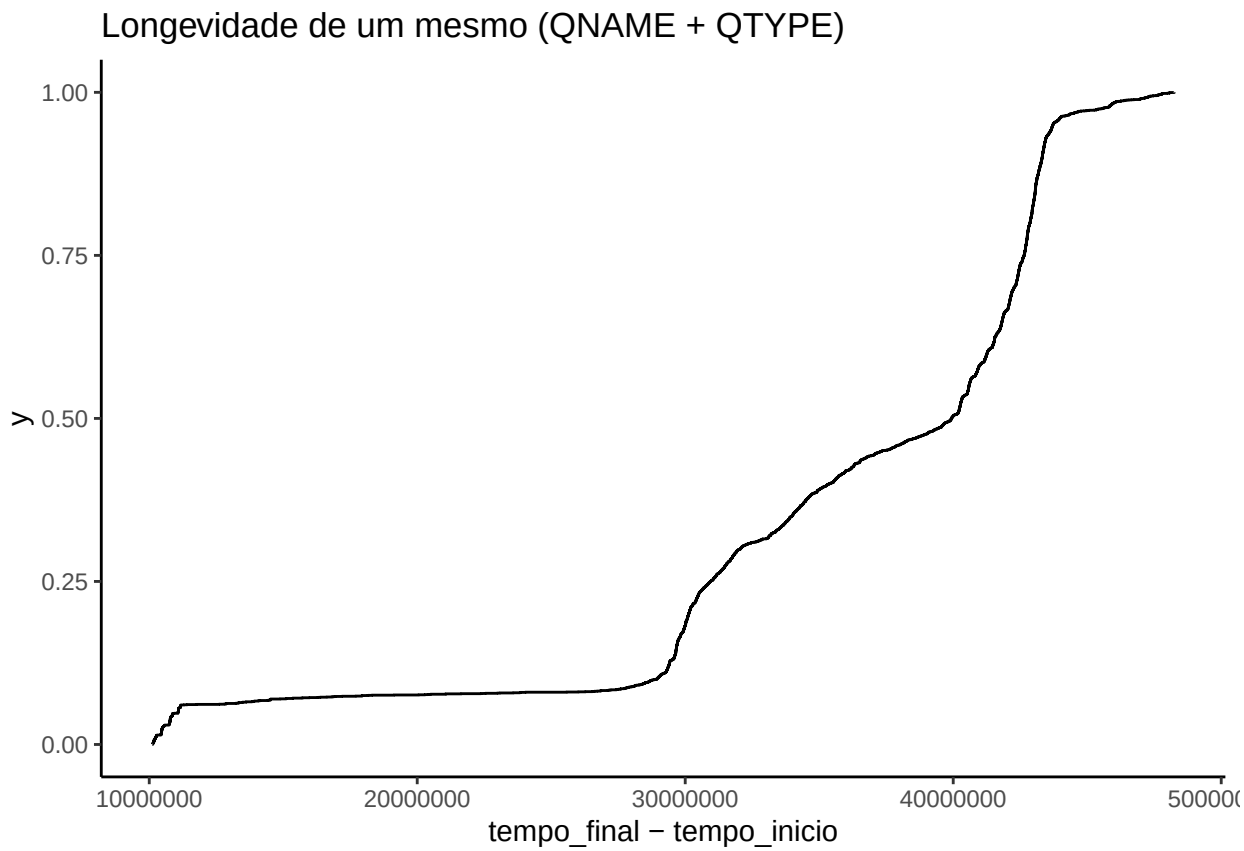
- Dados sobre o intervalo entre a primeira e a última aparição do (QNAME+QTYPE) removendo 30% dos valores máximos e mínimos
  - Mínimo 0 segundos
  - Máximo 18.6145 meses
  - Média 7.2913 minutos
  - Mediana 67 segundos
- Identificar qual é o percentil em que há essa mudança de tendência (próximo aos 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%     69.63%     71.03%        72%     72.04%     72.11%
##       70.0      745.6     3602.7     5457.6     7265.0     7434.6     7764.7
##        73%     73.97%        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 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)
  - Possívelmente 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] 0.4745
```

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

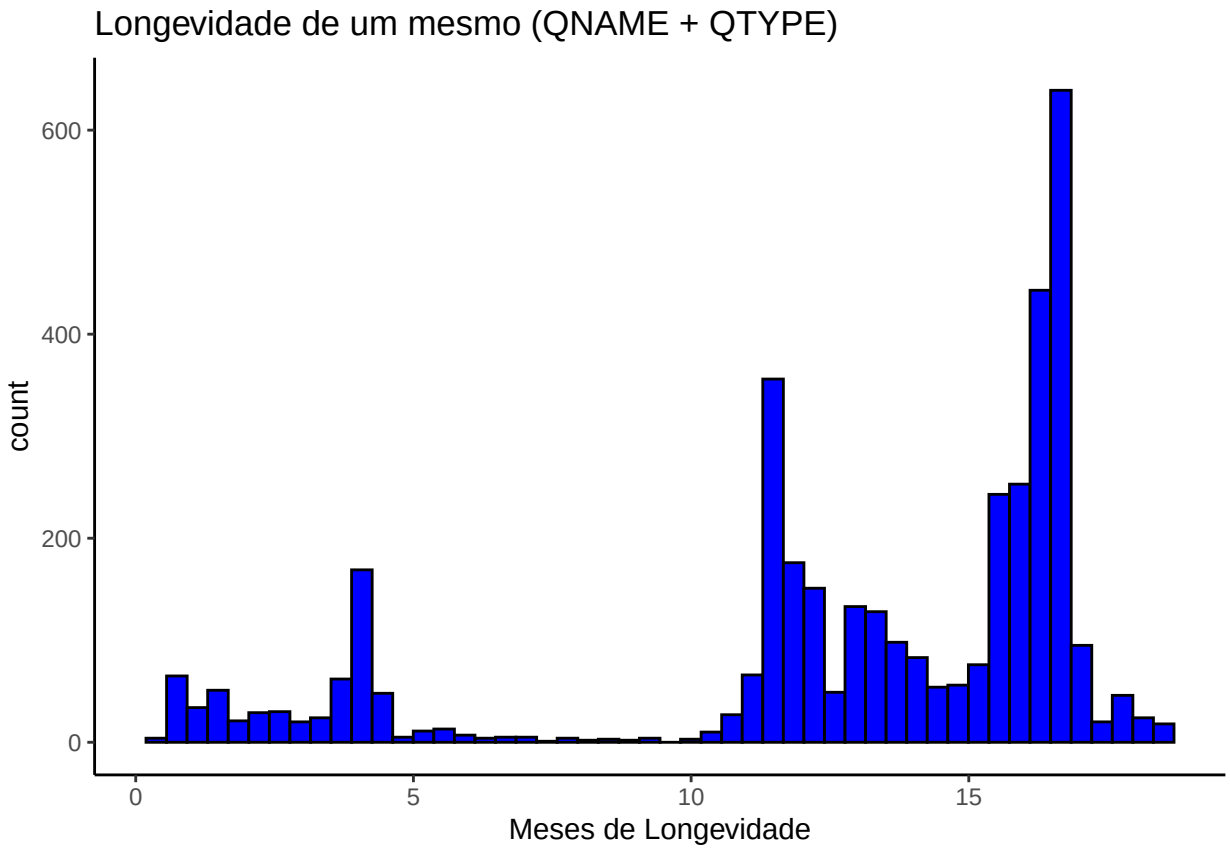
```
## [1] 17.66
```

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

```
## [1] 18.61
```

- 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()
```

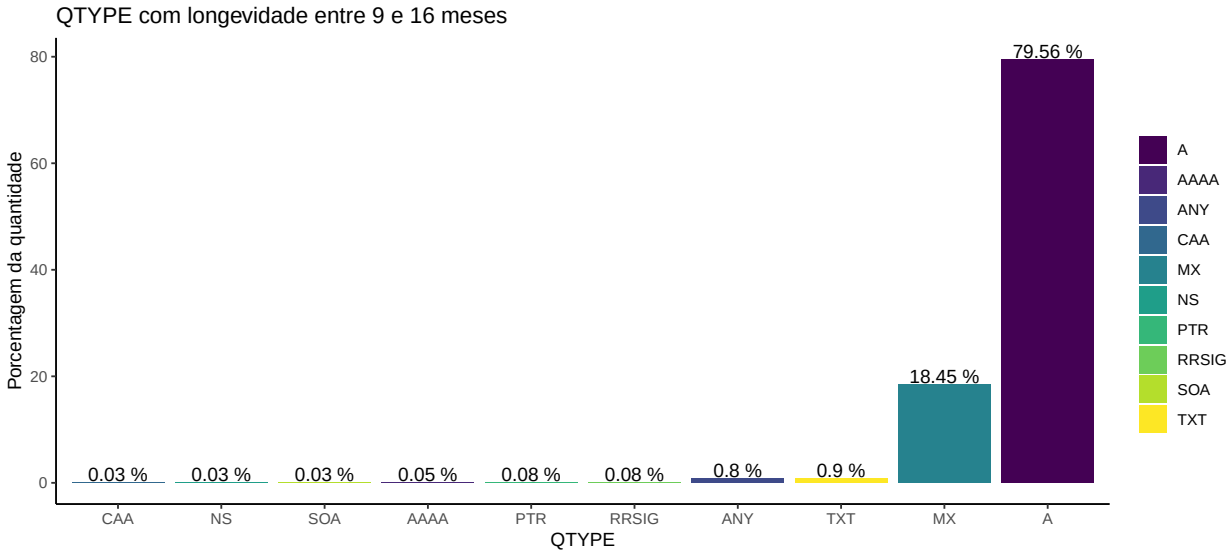



```
data_bigger_than_76 = data_grouped %>%
  filter(tempo_diff_secs > percentage_76_secs) %>%
  ungroup() %>%
  group_by(qtype) %>%
  summarise(qtype_quantity = n()) %>%
  arrange(desc(qtype_quantity))


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 agroupado 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_
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_fina

percentage_76_secs_qtype_A_group_qname %>%
  arrange(desc(tempo_diff_secs)) %>%
  head(10)
```

```
## # A tibble: 10 x 7
##    qname    qname_count sum_requests_pe~ tempo_inicio        tempo_final
##    <chr>          <int>            <int> <dttm>              <dttm>
##  1 200-19-~          98              148 2020-10-30 11:15:36 2022-05-09 21:23:22
##  2 a.gtld-~          84               89 2020-10-30 02:55:07 2022-05-09 06:50:12
##  3 public1~         527             1812 2020-11-04 09:47:54 2022-05-10 13:47:55
##  4 dnsscan~         111              116 2020-11-05 15:30:18 2022-05-09 00:23:03
##  5 amazon.~         454             2196 2020-11-10 15:16:10 2022-05-11 10:44:57
##  6 direct.~           4                4 2020-10-30 02:58:00 2022-04-27 10:32:50
##  7 whoami.~        5742             6121 2020-10-30 12:33:16 2022-04-27 03:14:30
##  8 www.bb.~          29               60 2020-11-05 15:04:15 2022-04-28 08:09:13
##  9 200-19-~         122              265 2020-11-22 11:21:53 2022-05-09 21:23:22
## 10 gmail.c~          73             1317 2020-11-24 10:26:30 2022-05-10 23:15:29
## # ... with 2 more variables: tempo_diff_secs <dbl>, tempo_diff <drtn>
```

- Uníca 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_pe~ tempo_inicio        tempo_final
##    <chr>          <int>            <int> <dttm>              <dttm>
##  1 www.ndn~        3357            39546 2021-11-06 10:47:11 2022-03-17 05:44:18
##  2 www.ac.~        1705            28843 2021-07-09 07:42:37 2022-01-19 21:38:20
##  3 www.ac.~         771            26210 2021-07-14 09:25:15 2022-01-19 21:38:18
##  4 admin.a~          60            21672 2020-12-28 05:54:53 2022-05-07 11:58:18
##  5 probe.i~        1108            13150 2022-02-21 13:17:16 2022-05-04 05:43:39
##  6 theguar~          62            12805 2021-02-07 10:19:59 2022-05-10 00:14:05
##  7 ftp.ebi~          47            12035 2020-12-25 00:28:07 2022-05-03 15:29:10
##  8 dji.gov~          63            11492 2020-12-27 01:14:12 2022-04-19 04:38:40
##  9 hotspot~          52            11366 2021-02-07 10:19:50 2022-05-10 15:44:37
## 10 2015ann~        4264            10967 2021-09-14 10:36:59 2022-05-09 14:14:18
## # ... 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(te
  mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_fin
  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> <dttm>
##  1    50265            660                    1487 2020-12-18 04:26:58
##  2    45810            323                     401 2020-12-18 12:30:00
##  3    28826            215                     282 2021-03-01 10:34:11
##  4    64206             99                     102 2021-02-05 13:23:20
##  5     4218             98                     110 2020-10-30 18:46:37
##  6     1337             86                     100 2020-10-30 02:55:07
##  7    44557             76                      83 2020-11-26 11:07:27
##  8    14602             64                     127 2021-10-23 22:49:59
##  9    16028             49                    3793 2020-11-10 02:29:52
## 10    36379             45                     142 2021-05-11 12:27:03
## # ... with 3 more variables: tempo_final <dttm>, 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
#    mutate(tempo_diff_secs = as.numeric(tempo_final - tempo_inicio, units="secs"), tempo_diff = tempo_
#    arrange(desc(tempo_diff_secs))
#
#  period_query_id_qname = rbind(period_query_id_qname, head(query_id_qname_frequency, N))
#}
```

Registros separados por trimestre

- Apenas verificando as porcentagens de QTYPE por trimestre

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

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

```
## # A tibble: 10 x 5
## # Groups:   year_period [7]
##    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       20222 A                         18404            27094             67.9
##  4       20212 ANY                       23182            39611             58.5
##  5       20213 A                         30519            53934             56.6
##  6       20221 A                         10103            18869             53.5
##  7       20214 A                         47906            98317             48.7
##  8       20214 ANY                       39673            98317             40.4
##  9       20221 ANY                        7217            18869             38.2
## 10       20212 A                         12326            39611             31.1
```

- 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

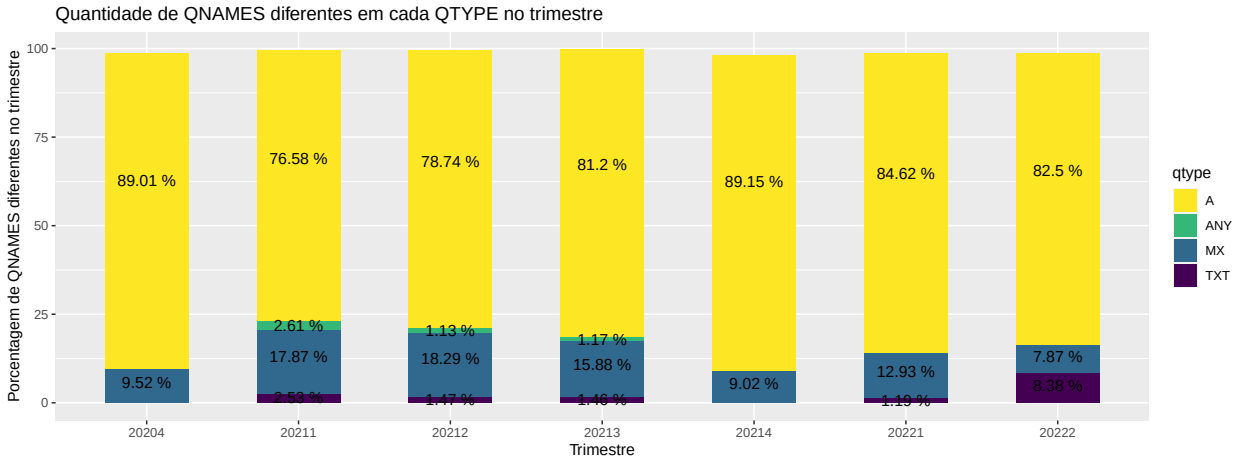
```
data_year_period_qtype_quantity = data_year_period %>%
  ungroup() %>%
  group_by(year_period, qtype) %>%
  summarise(qtype_quantity = n())
```

```
## `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