

dns__first

2022-03-05

R Markdown

```
library('RSQLite')
library('ggplot2')
library(DBI)
options("scipen"=100, "digits"=4)

db <- dbConnect(RSQLite::SQLite(), dbname="./dnstor_statistics_dns.sqlite")
dns_data <- dbSendQuery(db, "
  SELECT count(*) as countGrouped, year, period, 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
GROUP BY year_period, year, period, qname, qtype
ORDER BY quantity DESC;
")
dns_data_fetched <- fetch(dns_data)
#dns_data_fetched %>%
# filter(qtype == 0)
```

```
library(dplyr)
```

```
##
## 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
```

```
library(tibble)

dns_data.year_period.ungrouped <- group_split(dns_data_fetched, year_period)

N = 10
dns_data.topNconsultas <- head(dns_data.year_period.ungrouped[[1]], N)
```

```
dns_data.year_period.ungrouped.len = length(dns_data.year_period.ungrouped)

select(dns_data.topNconsultas, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 10 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1     20204 ANY    19005578 peacecorps.gov.
## 2     20204 ANY     816242 lavrov.in.
## 3     20204 ANY     779892 sl.
## 4     20204 ANY     652325 irs.gov.
## 5     20204 ANY     569411 fe18.ru.
## 6     20204 ANY      12296 .
## 7     20204 ANY     10248 isc.org.
## 8     20204 A       8467 20200328132334-cq9bm.ldd.sohu.com.
## 9     20204 RRSIG    6176 jp.
## 10    20204 A       4953 500940734da64dde863b257c9c12c03d.apigw.ap-southea~
```

```
select(head(dns_data.year_period.ungrouped[[2]], N), c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 10 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1     20211 ANY    32698124 peacecorps.gov.
## 2     20211 ANY     3032399 sl.
## 3     20211 ANY     2418859 isc.org.
## 4     20211 ANY     941083 fe18.ru.
## 5     20211 ANY     463904 wzb.eu.
## 6     20211 ANY     132970 .
## 7     20211 A       20998 mirrorlist.centos.org.
## 8     20211 A       10698 hotspot.accesscam.org.
## 9     20211 MX       8014 pwad.gov.ae.
## 10    20211 A       3882 theguardian.webredirect.org.
```

```
select(head(dns_data.year_period.ungrouped[[3]], N), c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 10 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1     20212 ANY    13183512 peacecorps.gov.
## 2     20212 ANY     1337802 sl.
## 3     20212 ANY     534815 irs.gov.
## 4     20212 ANY     220674 isc.org.
## 5     20212 ANY     124579 fe18.ru.
## 6     20212 ANY      90999 .
## 7     20212 MX      21895 dpc.ae.
## 8     20212 ANY     11229 hcc.nl.
## 9     20212 A      10965 dji.gov.ae.
## 10    20212 A       9144 emaratalyounm.com.
```

```
select(head(dns_data.year_period.ungrouped[[4]], N), c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 10 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1    20213 RRSIG    324789 pizzaseo.com.
## 2    20213 ANY     178363 sl.
## 3    20213 ANY     165932 .
## 4    20213 A        5925 www.ac.my.blastodermic-swimmable.info.
## 5    20213 A        5291 tmall.com.
## 6    20213 A        4848 www.ac.my.superability-kooka.info.
## 7    20213 A        4655 2015annualreport.bloomberg.org.
## 8    20213 A        2794 lpnkuearwljpqwbwz.tmall.com.
## 9    20213 MX        1915 rt.com.
## 10   20213 MX        1888 nawahprogram.ae.
```

```
select(head(dns_data.year_period.ungrouped[[5]], N), c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 10 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1    20214 ANY    4844082 peacecorps.gov.
## 2    20214 ANY     620249 sl.
## 3    20214 A       19541 www.ac.my.blastodermic-swimmable.info.
## 4    20214 A       17848 www.ac.my.superability-kooka.info.
## 5    20214 A       13595 www.ndnslab.com.
## 6    20214 ANY      11073 .
## 7    20214 RRSIG     8499 pizzaseo.com.
## 8    20214 MX       6670 nih.gov.
## 9    20214 A       5932 2015annualreport.bloomberg.org.
## 10   20214 MX       4680 nawahprogram.ae.
```

```
select(head(dns_data.year_period.ungrouped[[6]], N), c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 10 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1    20221 ANY    2614699 peacecorps.gov.
## 2    20221 A       21200 admin.asry.net.
## 3    20221 ANY     19737 sl.
## 4    20221 A       18629 www.ndnslab.com.
## 5    20221 A       11635 ftp.ebisb.com.
## 6    20221 MX       7821 bankfab.com.
## 7    20221 A       6091 vpn.qatarsteel.com.qa.
## 8    20221 MX       6025 zayed.org.ae.
## 9    20221 A       5766 moi.gov.kw.
## 10   20221 MX       5077 mopa.ae.
```

```
for (i in c(2:dns_data.year_period.ungrouped.len)) {
  dns_data.topNconsultas <- rbind(dns_data.topNconsultas, head(dns_data.year_period.ungrouped[[i]], N))
}
```

```
#print(select(dns_data.topNconsultas, c('year_period', 'qtype', 'quantity', 'qname')), nrow = length(dn
```

```
## ----- ataques que repetem o query_id -----
top_10_repeated_query_data <-dbSendQuery(db, "
  SELECT count(*) as qnt_repeat_query_id, year, period, count, query_id, qname, qtype
  FROM DNS_ANALYSIS
  JOIN DNS_ANALYSIS_QUESTION
  ON DNS_ANALYSIS.id = DNS_ANALYSIS_QUESTION.dns_analysis_id
GROUP BY query_id
  HAVING qnt_repeat_query_id > 1
ORDER BY qnt_repeat_query_id desc
  LIMIT 10;
")
```

```
## Warning: Closing open result set, pending rows
```

```
top_10_repeated_query <- fetch(top_10_repeated_query_data)

qnames = top_10_repeated_query['qname']

top_10_repeated_query.1 = head(dns_data.year_period.ungrouped[[1]], N) %>%
  filter(qname %in% top_10_repeated_query$qname)

top_10_repeated_query.2 = head(dns_data.year_period.ungrouped[[2]], N) %>%
  filter(qname %in% top_10_repeated_query$qname)

top_10_repeated_query.3 = head(dns_data.year_period.ungrouped[[3]], N) %>%
  filter(qname %in% top_10_repeated_query$qname)

top_10_repeated_query.4 = head(dns_data.year_period.ungrouped[[4]], N) %>%
  filter(qname %in% top_10_repeated_query$qname)

top_10_repeated_query.5 = head(dns_data.year_period.ungrouped[[5]], N) %>%
  filter(qname %in% top_10_repeated_query$qname)

top_10_repeated_query.6 = head(dns_data.year_period.ungrouped[[6]], N) %>%
  filter(qname %in% top_10_repeated_query$qname)

top_10_repeated_query = rbind(top_10_repeated_query.1, top_10_repeated_query.2, top_10_repeated_query.3,
                              top_10_repeated_query.4, top_10_repeated_query.5, top_10_repeated_query.6)

top_10_repeated_query %>%
  arrange(desc(quantity))
```

```
## # A tibble: 9 x 7
```

```
##   countGrouped year period year_period qname          qtype quantity
##   <int> <int>  <int>      <int> <chr>          <chr>      <int>
## 1      35320  2021      1      20211 peacecorps.gov. ANY      32698124
## 2      23891  2020      4      20204 peacecorps.gov. ANY      19005578
```

```
## 3      18311 2021      2      20212 peacecorps.gov. ANY 13183512
## 4      29963 2021      4      20214 peacecorps.gov. ANY 4844082
## 5       5876 2022      1      20221 peacecorps.gov. ANY 2614699
## 6      45508 2020      4      20204 irs.gov.      ANY 652325
## 7        11 2021      2      20212 irs.gov.      ANY 534815
## 8       6256 2021      3      20213 pizzaseo.com. RRSIG 324789
## 9        118 2021      4      20214 pizzaseo.com. RRSIG 8499
```

```
select(top_10_repeated_query.1, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 2 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1      20204 ANY    19005578 peacecorps.gov.
## 2      20204 ANY      652325 irs.gov.
```

```
select(top_10_repeated_query.2, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 1 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1      20211 ANY    32698124 peacecorps.gov.
```

```
select(top_10_repeated_query.3, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 2 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1      20212 ANY    13183512 peacecorps.gov.
## 2      20212 ANY      534815 irs.gov.
```

```
select(top_10_repeated_query.4, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 1 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1      20213 RRSIG    324789 pizzaseo.com.
```

```
select(top_10_repeated_query.5, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 2 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1      20214 ANY    4844082 peacecorps.gov.
## 2      20214 RRSIG      8499 pizzaseo.com.
```

```
select(top_10_repeated_query.6, c('year_period', 'qtype', 'quantity', 'qname'))
```

```
## # A tibble: 1 x 4
##   year_period qtype quantity qname
##   <int> <chr>   <int> <chr>
## 1      20221 ANY    2614699 peacecorps.gov.
```

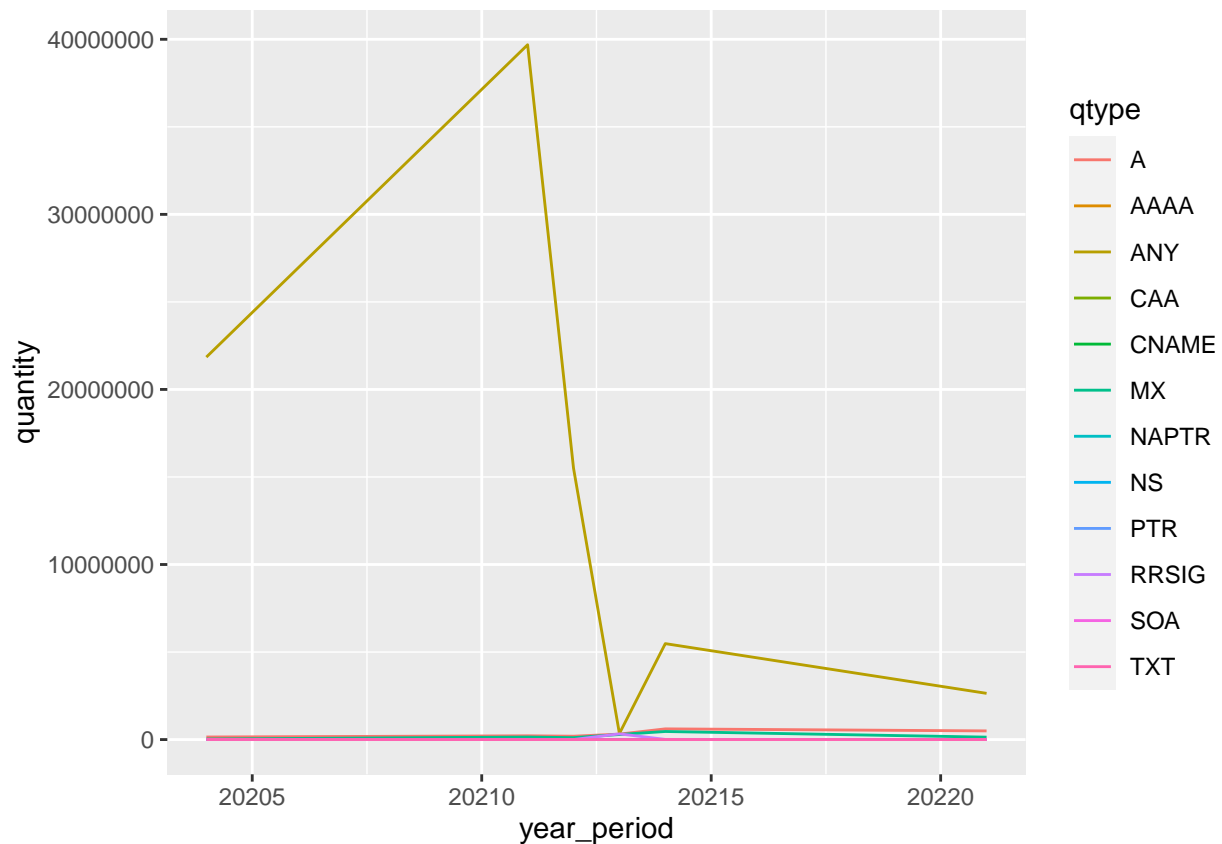
```
## ----- Quantos ataques com cada tipo de qtype foi utilizado, por trimestre ? -----
#dns_data_fetched

dns_data_fetched.quarter_type_quantity = select(dns_data_fetched, c('year_period', 'qtype', 'quantity'))

dns_data_fetched.sum_attacks_quarterly = dns_data_fetched.quarter_type_quantity %>%
  group_by(qtype, year_period) %>%
  summarise(quantity = sum(quantity))

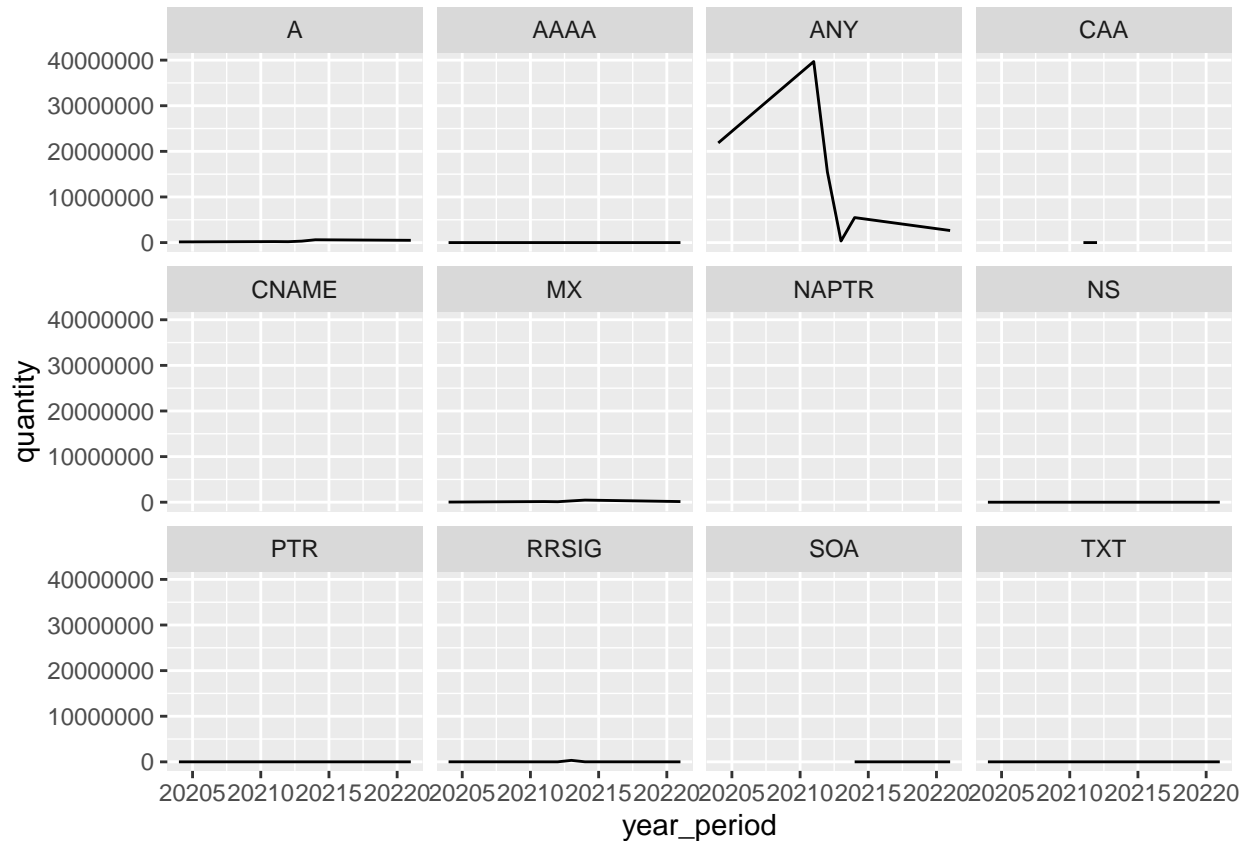
## 'summarise()' has grouped output by 'qtype'. You can override using the
## '.groups' argument.

dns_data_fetched.sum_attacks_quarterly %>%
# mutate(year_period=as.factor(year_period)) %>%
  ggplot(aes(x = year_period, y = quantity, color = qtype)) +
  geom_line()
```



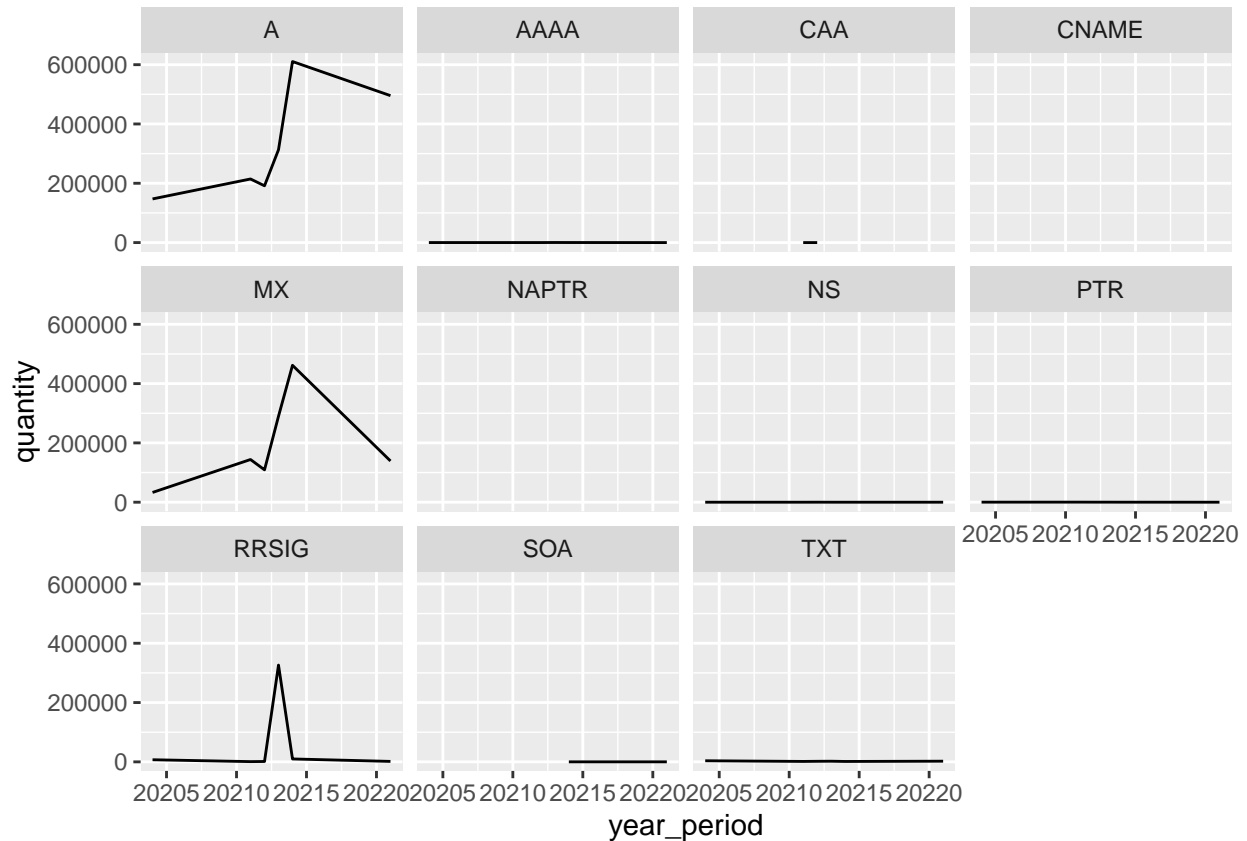
```
ggplot(data = dns_data_fetched.sum_attacks_quarterly, aes(x = year_period, y = quantity)) +
  geom_line() +
  facet_wrap(facets = vars(qtype))
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



```
dns_data_fetched.sum_attacks_quarterly %>%
  filter(qtype != "ANY") %>%
  ggplot(aes(x = year_period, y = quantity)) +
    geom_line() +
    facet_wrap(facets = vars(qtype))
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



```
# ----- quantity with percentage

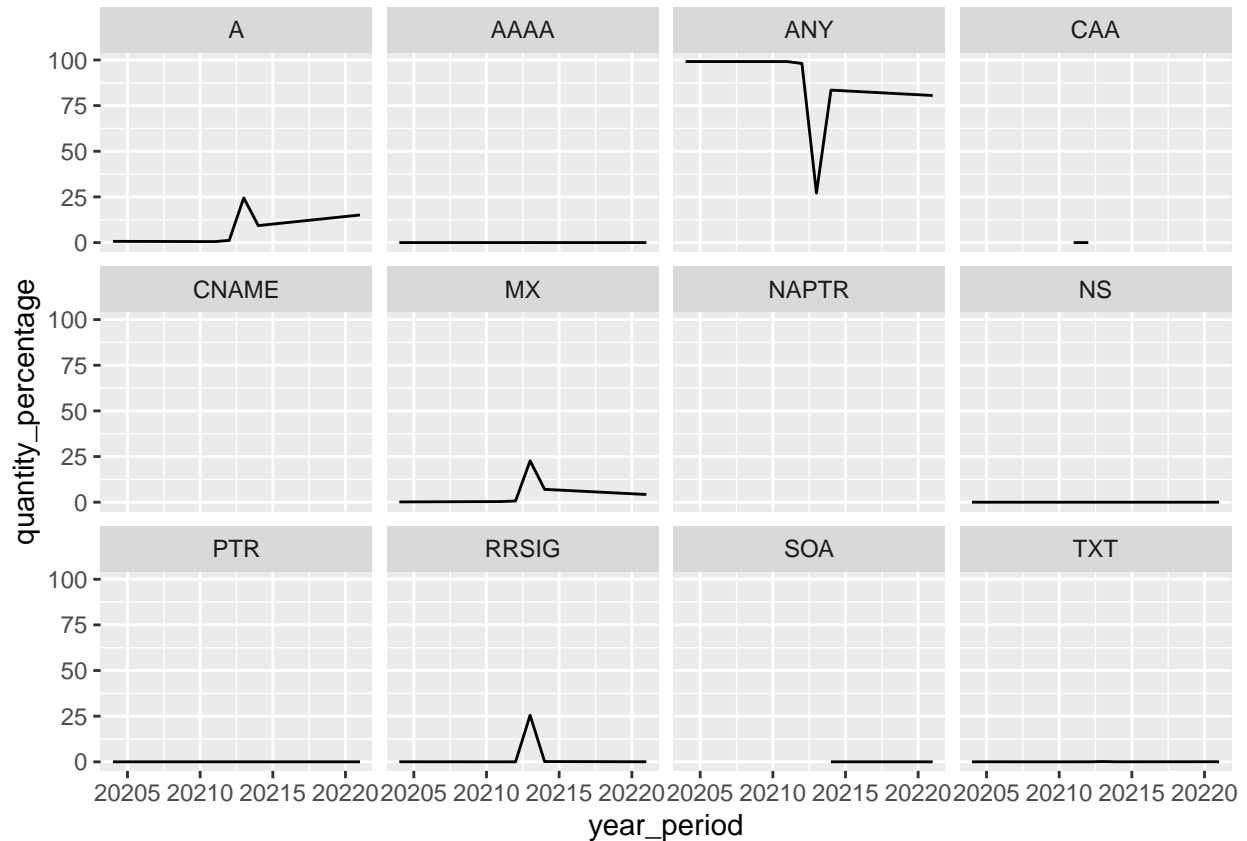
dns_data_fetched.sum_attacks_quarterly.sum_period_quantity = dns_data_fetched.sum_attacks_quarterly %>%
  group_by(year_period) %>%
  summarise(sum_period_quantity = sum(quantity), qtype=qtype, quantity=quantity)
```

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

```
dns_data_fetched.sum_attacks_quarterly.sum_period_quantity['quantity_percentage'] = (dns_data_fetched.s

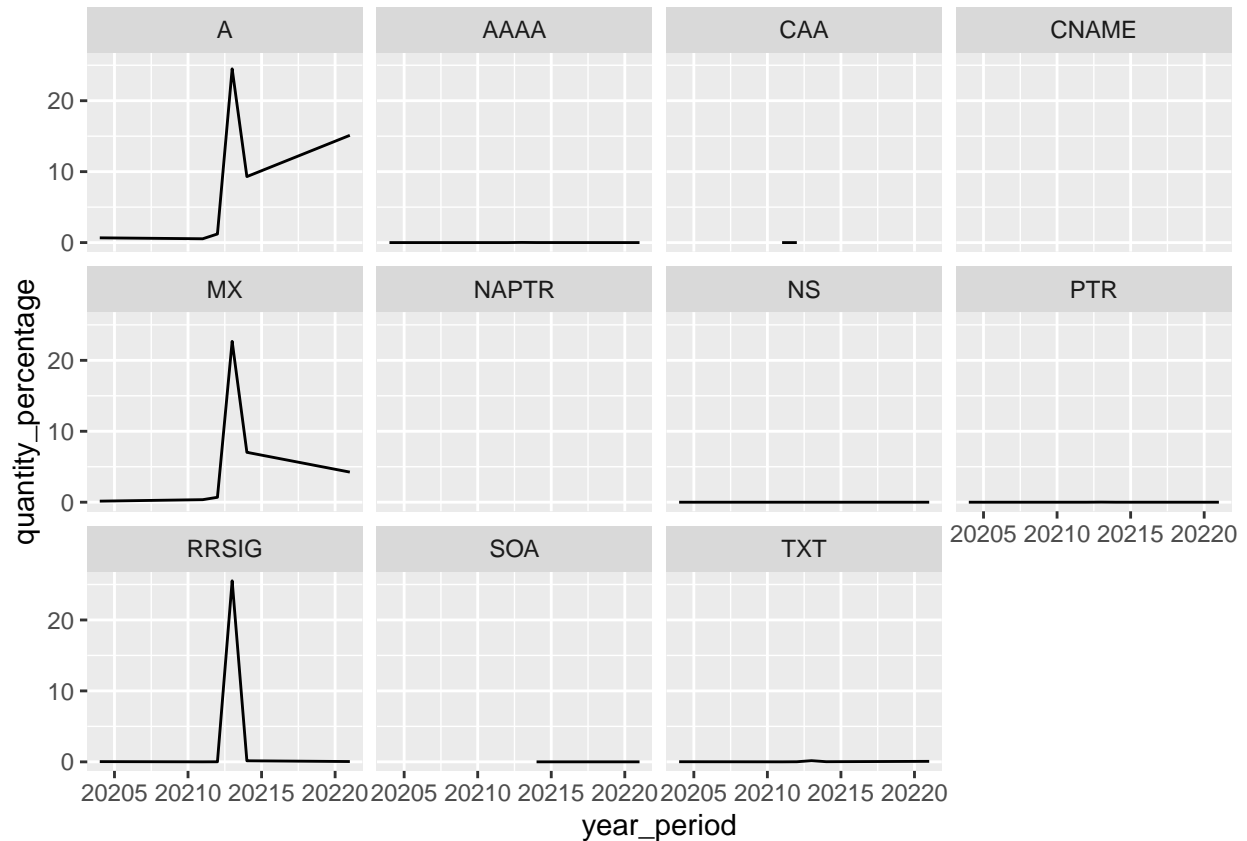
dns_data_fetched.sum_attacks_quarterly.sum_period_quantity %>%
  ggplot(aes(x = year_period, y = quantity_percentage)) +
  geom_line() +
  facet_wrap(facets = vars(qtype))
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

```
dns_data_fetched.sum_attacks_quarterly.sum_period_quantity %>%
  filter(qtype != "ANY") %>%
  ggplot(aes(x = year_period, y = quantity_percentage)) +
    geom_line() +
    facet_wrap(facets = vars(qtype))
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



```
# ----- filter any

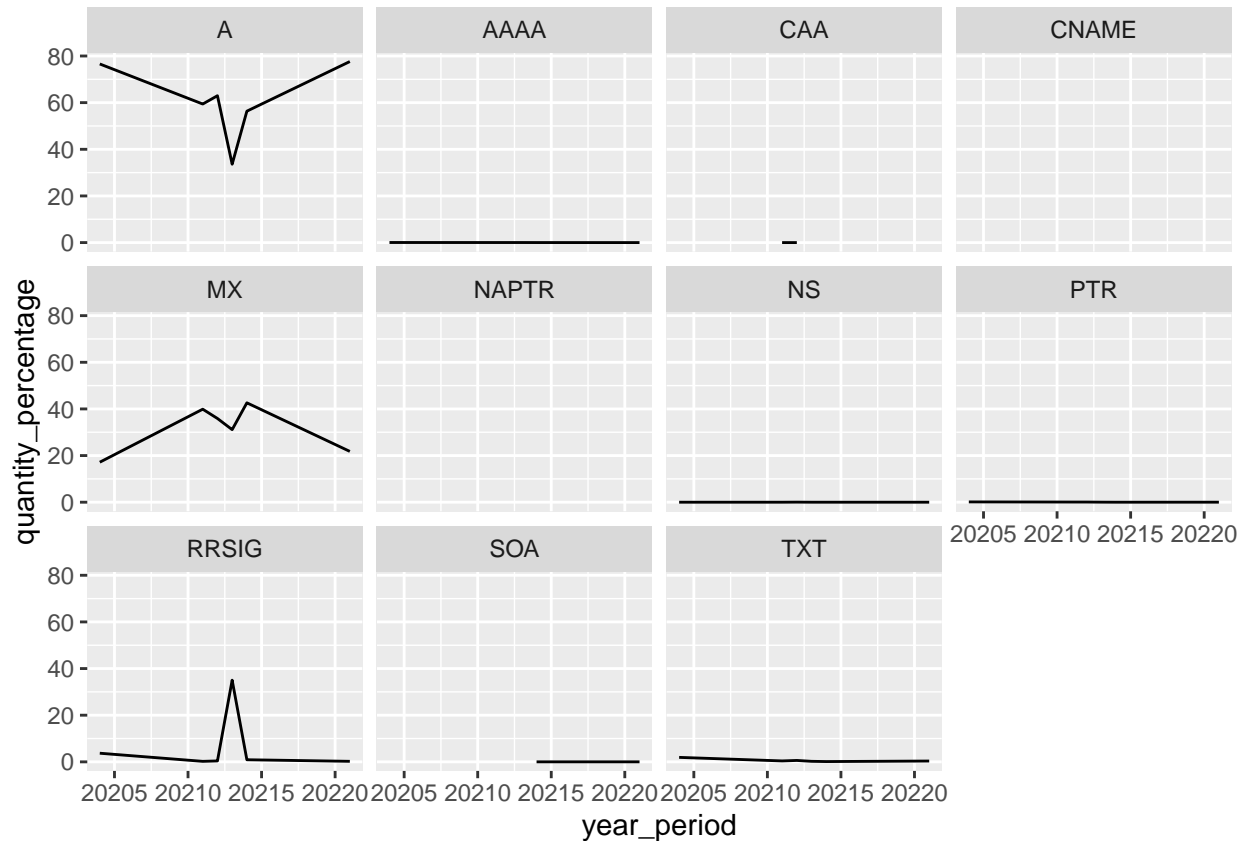
dns_data_fetched.sum_attacks_quarterly.sum_period_quantity.filter_any = dns_data_fetched.sum_attacks_quarterly.sum_period_quantity
  group_by(year_period) %>%
  filter(qtype != "ANY") %>%
  summarise(sum_period_quantity = sum(quantity), qtype=qtype, quantity=quantity)
```

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

```
dns_data_fetched.sum_attacks_quarterly.sum_period_quantity.filter_any['quantity_percentage'] = (dns_data_fetched.sum_attacks_quarterly.sum_period_quantity.filter_any['quantity_percentage'] * 100)

dns_data_fetched.sum_attacks_quarterly.sum_period_quantity.filter_any %>%
  ggplot(aes(x = year_period, y = quantity_percentage)) +
  geom_line() +
  facet_wrap(facets = vars(qtype))
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



```
# Tiago
# - A e MX devem ser olhados junto com o ANY pra ver se existe alguma relação com esse crescimento
# - RRSIG tem um pico legal (descobrir qual ataque/relação pra tentar entender seria interessante)
# - todos os outros qtype deveriam ser gerados em outro grafico pra ver se o padrão d RRSIG n aparece t

# ----- Quantos qtypes novos aparecem em cada trimestre -----
# > Diferenças percentuais são mais relevantes que absolutas

quarter_qtype_aux = dns_data.year_period.ungrouped[[1]] %>%
  group_by(qtype) %>%
  summarise(quantity = sum(quantity))

#quarter_qtype_2 = dns_data.year_period.ungrouped[[2]] %>%
# group_by(qtype) %>%
# summarise(quantity = sum(quantity))

#quarter_qtype_2
#merged = merge(x = quarter_qtype_aux, y = quarter_qtype_2, by = "qtype", all = TRUE)
#merged.new_quantity = merged$quantity.x - merged$quantity.y
#merged

quarter_new_qtype = data.frame()
```

```

for (i in c(2:dns_data.year_period.ungrouped.len)) {
  quarter_qtype = dns_data.year_period.ungrouped[[i]] %>%
    group_by(qtype) %>%
    summarise(quantity = sum(quantity))

  merged = merge(x = quarter_qtype_aux, y = quarter_qtype, by = "qtype", all = TRUE)
  merged.new_quantity = merged$quantity.x - merged$quantity.y

  perio_to_period = paste(head(dns_data.year_period.ungrouped[[i - 1]]['year'], 1), '.', head(dns_data.
  quarter_new_qtype <- rbind(quarter_new_qtype, data.frame(quarter_to_quarter=perio_to_period, merged$quantity.y))

  quarter_qtype_aux = quarter_qtype
}

#quarter_new_qtype
head(na.omit(quarter_new_qtype[order(-quarter_new_qtype$quantity_percentage),]))

```

```

##      quarter_to_quarter merged.qtype sum_quantity quantity_percentage
## 28 2021 . 2 -> 2021 . 3      RRSIG      325120      26803.0
## 17 2021 . 1 -> 2021 . 2         NS         119      2975.0
## 32 2021 . 3 -> 2021 . 4         ANY     5133467      1480.4
## 22 2021 . 2 -> 2021 . 3       AAAA         195       367.9
## 6  2020 . 4 -> 2021 . 1         MX     111066      336.9
## 43 2021 . 4 -> 2022 . 1         NS          2      200.0
##      merged.quantity.x merged.quantity.y
## 28              1213          326333
## 17                 4             123
## 32             346754          5480221
## 22                 53             248
## 6              32964          144030
## 43                 1              3

```

----- Quantos qname novos aparecem em cada trimestre -----

```

quarter_qname_aux = dns_data.year_period.ungrouped[[1]] %>%
  group_by(qname) %>%
  summarise(quantity = sum(quantity))

quarter_new_qname = data.frame()
for (i in c(2:dns_data.year_period.ungrouped.len)) {
  quarter_qname = dns_data.year_period.ungrouped[[i]] %>%
    group_by(qname) %>%
    summarise(quantity = sum(quantity))

  merged = merge(x = quarter_qname_aux, y = quarter_qname, by = "qname", all = TRUE)
  merged.new_quantity = merged$quantity.x - merged$quantity.y

  period_to_period = paste(head(dns_data.year_period.ungrouped[[i - 1]]['year'], 1), '.', head(dns_data.
  quarter_new_qname <- rbind(quarter_new_qname, data.frame(quarter_to_quarter=period_to_period, merged$quantity.y))

  quarter_qname_aux = quarter_qname
}

```

```
#quarter_new_qname
head(na.omit(quarter_new_qname[-order(quarter_new_qname$quantity_percentage_diff),]))
```

```
## [1] quarter_to_quarter      merged.qname      sum_quantity
## [4] quantity_percentage_diff merged.quantity.x    merged.quantity.y
## <0 rows> (or 0-length row.names)
```

```
# @todo
#1- olhar a longo prazo, o timelapse dos qnames
#2- qual a frequencia d qnames novos nesses períodos
# 2.1 olhar em detalhes as variações dos qnames (pq geralmente eles acabam sendo um grupo)
```

```
# Vale um gráfico de barras (dois, um agrupado e outro empilhado) da porcentagem de QTYPES por período
# https://www.data-to-viz.com/graph/barplot.html
# Libraries
library(viridis)
```

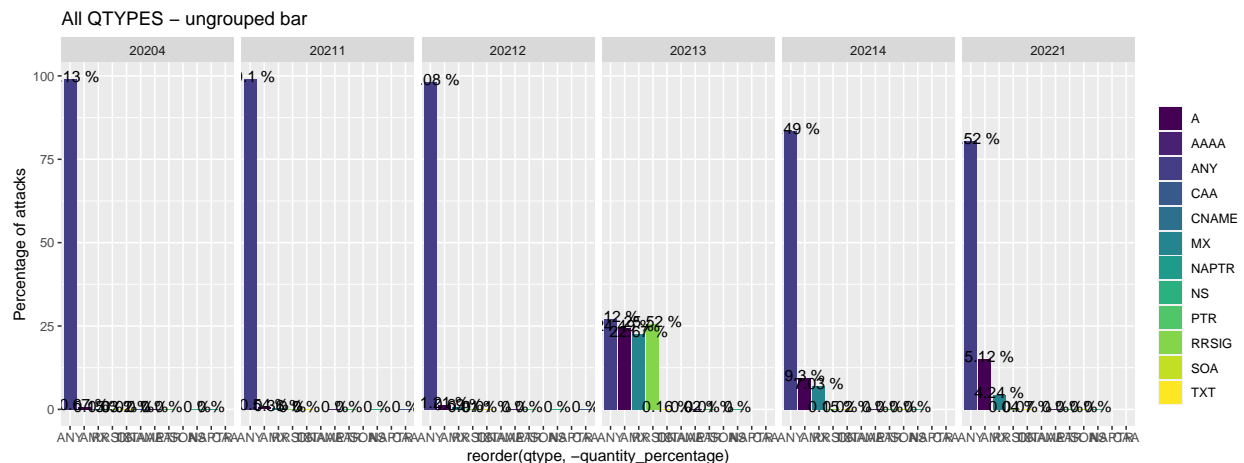
```
## Loading required package: viridisLite
```

```
dns_data_fetched.sum_attacks_quarterly.sum_period = dns_data_fetched.sum_attacks_quarterly %>%
  group_by(year_period) %>%
  summarise(period_quantity = sum(quantity), qtype=qtype, quantity=quantity)
```

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

```
dns_data_fetched.sum_attacks_quarterly.sum_period['quantity_percentage'] = (dns_data_fetched.sum_attacks_quarterly.sum_period$quantity_percentage / dns_data_fetched.sum_attacks_quarterly.sum_period$period_quantity) * 100
```

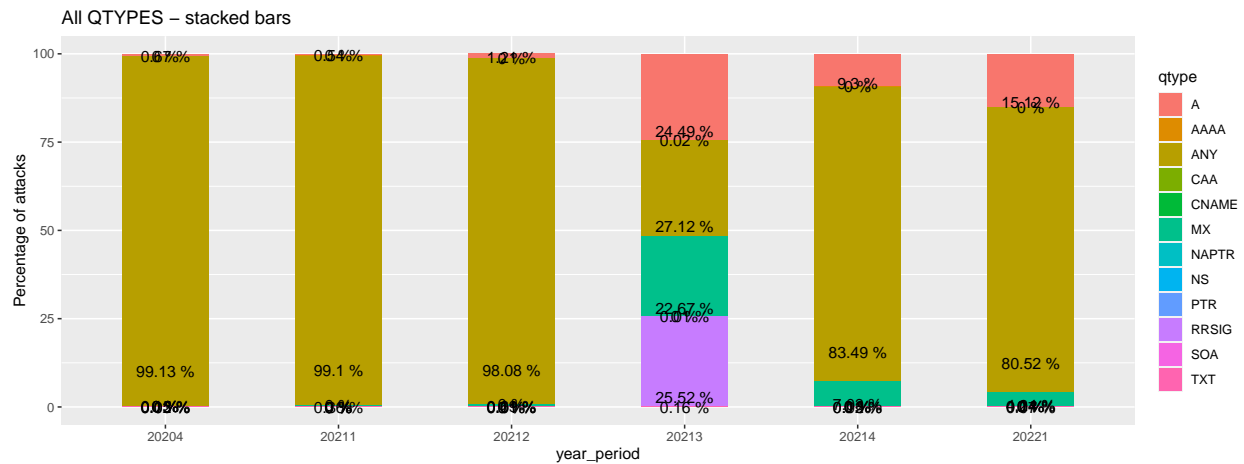
```
dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot(aes(x=reorder(qtype, -quantity_percentage), y=quantity_percentage, fill=qtype)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_viridis(discrete=TRUE, name="") +
  geom_text(aes(label = paste(round(quantity_percentage, 2), "%"), vjust = +0.25, ) +
  facet_grid(~year_period) +
  ylab("Percentage of attacks") +
  ggtitle("All QTYPES - ungrouped bar")
```



```

dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot( aes(x=year_period, y=quantity_percentage, fill=qtype)) +
    geom_bar(stat="identity", width = 0.5) +
    geom_text(aes(label = paste(round(quantity_percentage, 2), "%")), position = position_stack(vjust =
#scale_fill_viridis(discrete=TRUE, name="") +
  ylab("Percentage of attacks") +
  ggtitle("All QTYPES - stacked bars")

```

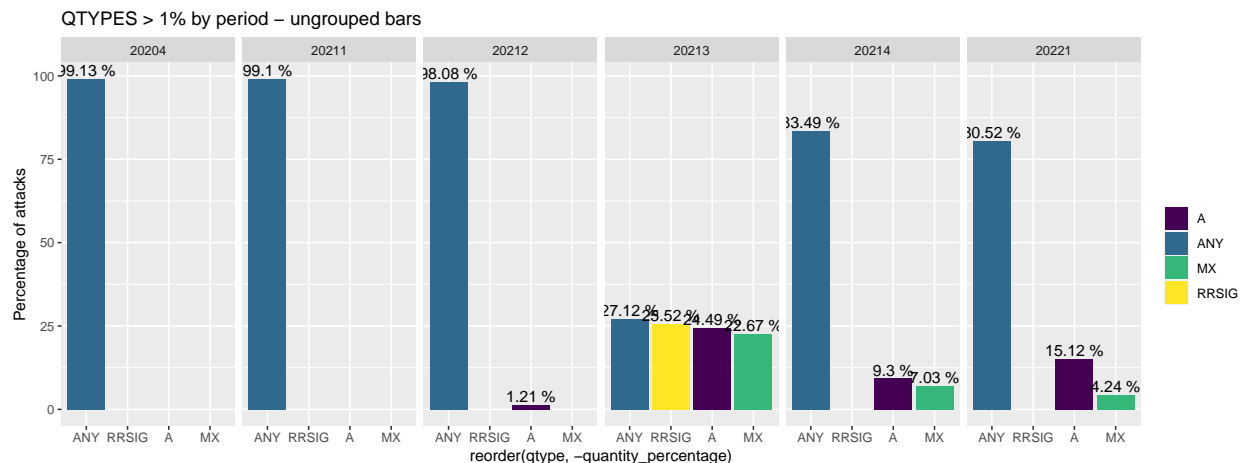


Filter data using qtype quantity percentage bigger than 1

```

dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  filter(quantity_percentage > 1) %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot( aes(x=reorder(qtype, -quantity_percentage), y=quantity_percentage, fill=qtype)) +
    geom_bar(stat="identity", position="dodge") +
    geom_text(aes(label = paste(round(quantity_percentage, 2), "%")), vjust = -0.25) +
    facet_grid(~year_period) +
    scale_fill_viridis(discrete=TRUE, name="") +
    ylab("Percentage of attacks") +
    ggtitle("QTYPES > 1% by period - ungrouped bars")

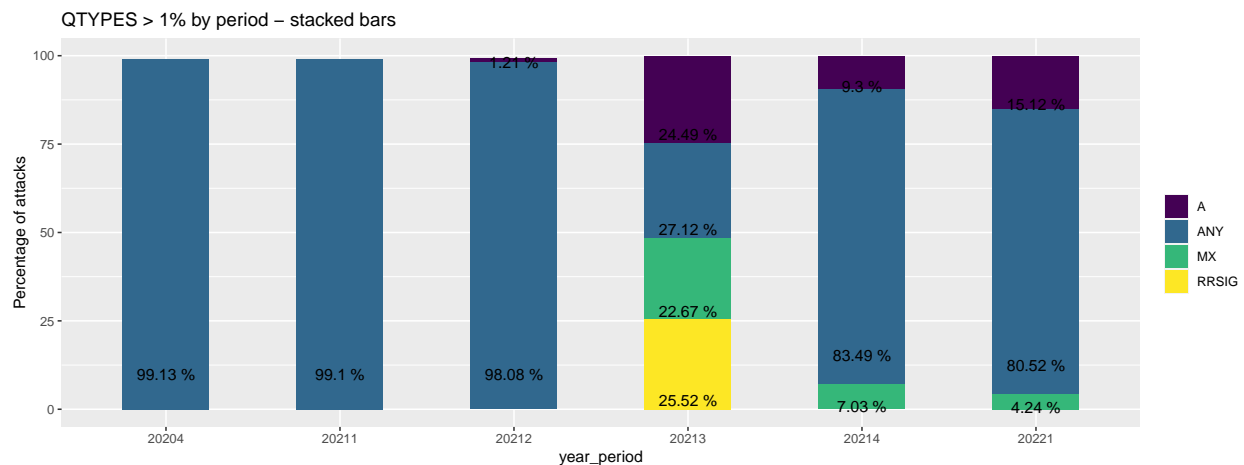
```



```

dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  filter(quantity_percentage > 1) %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot( aes(x=year_period, y=quantity_percentage, fill=qtype)) +
    geom_bar(stat="identity", width = 0.5) +
    geom_text(aes(label = paste(round(quantity_percentage, 2), "%")), position = position_stack(vjust =
    scale_fill_viridis(discrete=TRUE, name="") +
    ylab("Percentage of attacks") +
    ggtitle("QTYPES > 1% by period - stacked bars")

```



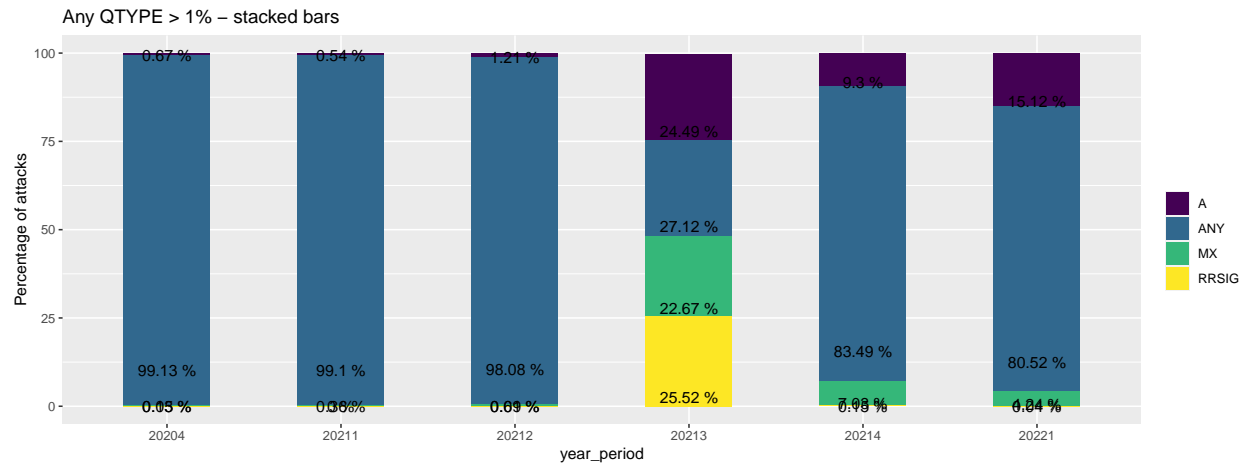
```

#dns_data_fetched.sum_attacks_quarterly.sum_period
dns_data_fetched.sum_attacks_quarterly.sum_period.relevant = dns_data_fetched.sum_attacks_quarterly.sum
  filter(quantity_percentage > 1)

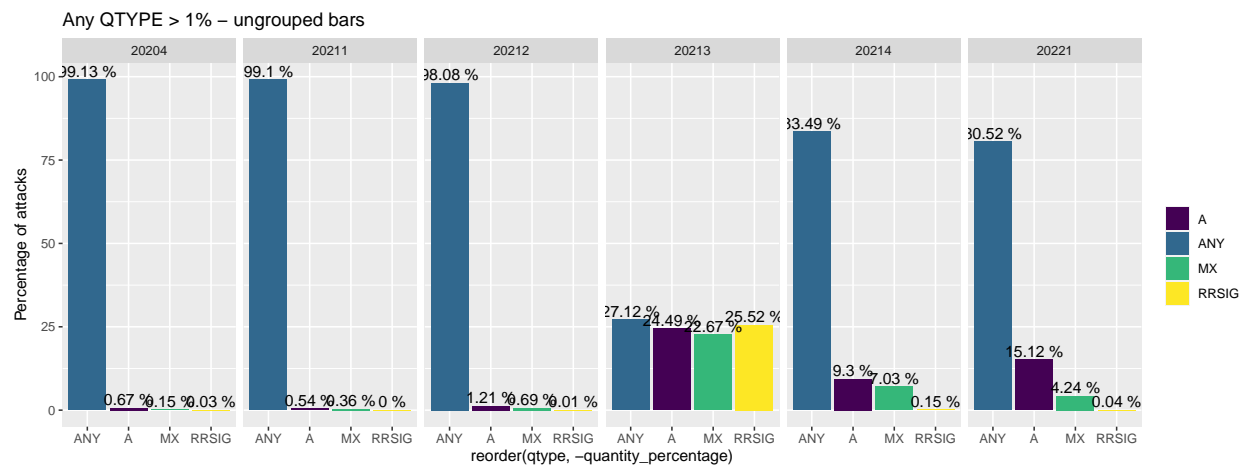
#dns_data_fetched.sum_attacks_quarterly.sum_period.relevant$qtype
qtypes_bigger_1 = dns_data_fetched.sum_attacks_quarterly.sum_period.relevant$qtype[!duplicated(dns_data
#qtypes_bigger_1

dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  filter(qtype %in% qtypes_bigger_1) %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot( aes(x=year_period, y=quantity_percentage, fill=qtype)) +
    geom_bar(stat="identity", width = 0.5) +
    geom_text(aes(label = paste(round(quantity_percentage, 2), "%")), position = position_stack(vjust =
    scale_fill_viridis(discrete=TRUE, name="") +
    ylab("Percentage of attacks") +
    ggtitle("Any QTYPE > 1% - stacked bars")

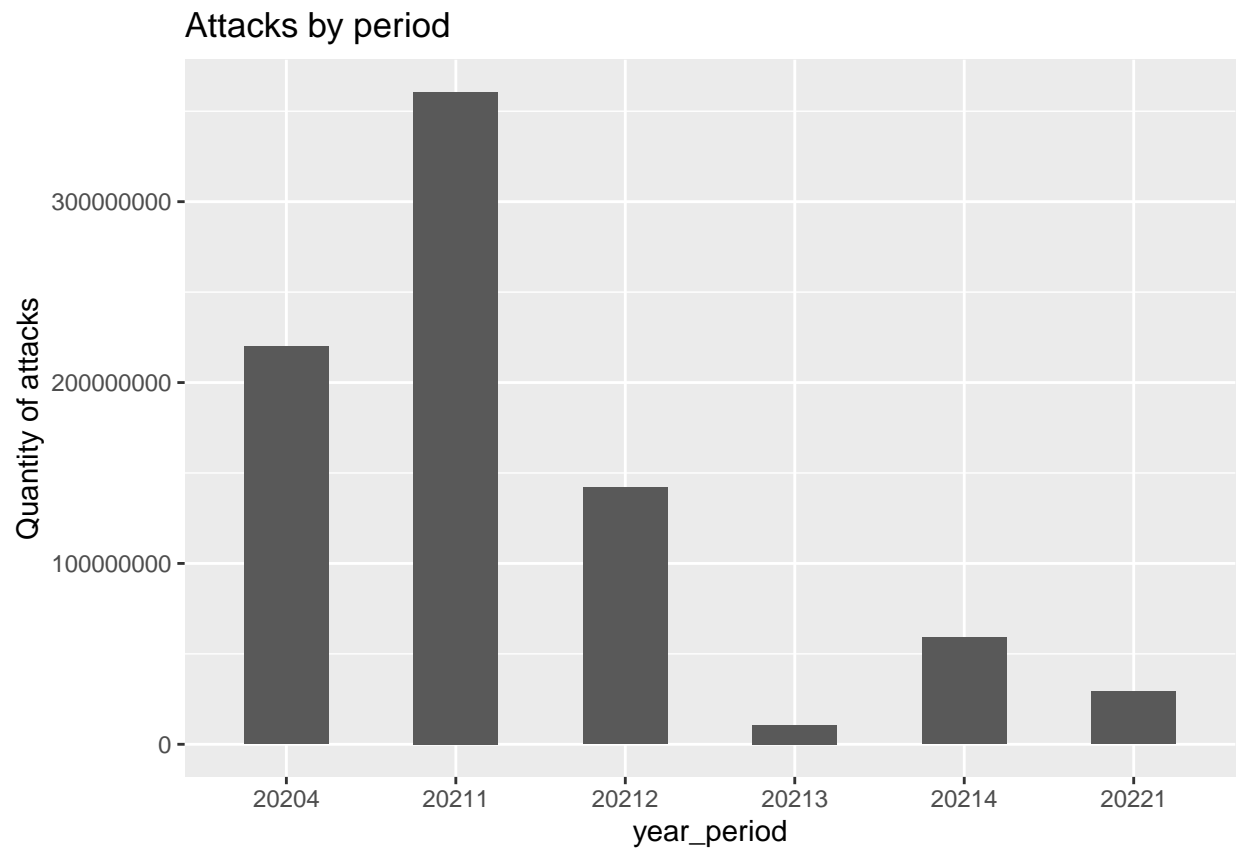
```



```
dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  filter(qtype %in% qtypes_bigger_1) %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot( aes(x=reorder(qtype, -quantity_percentage), y=quantity_percentage, fill=qtype)) +
    geom_bar(stat="identity", position="dodge") +
    geom_text(aes(label = paste(round(quantity_percentage, 2), "%"), vjust = -0.25) +
    facet_grid(~year_period) +
    scale_fill_viridis(discrete=TRUE, name="") +
    ylab("Percentage of attacks") +
    ggtitle("Any QTYPE > 1% – ungrouped bars")
```



```
dns_data_fetched.sum_attacks_quarterly.sum_period %>%
  mutate(year_period=as.factor(year_period)) %>%
  ggplot( aes(x=year_period, y=period_quantity)) +
    geom_bar(stat="identity", width = 0.5) +
    scale_fill_viridis(discrete=TRUE, name="") +
    ylab("Quantity of attacks") +
    ggtitle("Attacks by period")
```

```
# if each line on db were a request  
#dns_data_fetched. quarter_type_count.grouped_qtype_period %>%  
# mutate(year_period=as.factor(year_period)) %>%  
# ggplot( aes(x=year_period, y=count)) +  
#   geom_bar(stat="identity", width = 0.5) +  
#   scale_fill_viridis(discrete=TRUE, name="") +  
#   ylab("Quantity of request") +  
#   ggtitle("Request by period")
```