Superimposing evolutionary changes upon case trends in Canada

Supplementary Material: Endemic does not mean constant as SARS-CoV-2 continues to evolve

Sarah P. Otto, Ailene MacPherson, & Caroline Colijn

Evolution (2024)

Date of data download: September 23, 2023

---

duotang: Lineage evolution from December 1 2021 - August 31 2022

```
####################
#Ran duotang in RStudio, adjusting dates and lineages of interest to estimate the selection coefficients and lineage frequencies for different periods of time.
#Version : 22 September 2023
#Obtained from https://github.com/CoVaRR-NET/duotang
####################

####################
#Notes to run duotang.Rmd in RStudio :
#Set directories and dates needed to run duotang.Rmd
####################
setwd ("~/Dropbox/COVID19/CoVaRRNet_Pillar6_Notebook")
params <- list ()
params$datadir <-"data_needed/"
params$datestamp <-"2023-09-02"

#Desired date range
mystartdate <-“2021-12-01”
myenddate <-“2022-08-31”
params$datestamp <- myenddate

#Enter duotang.Rmd through “meta$month” then adjust meta to trim data after the desired date
```

```
 meta <- meta %>% filter(
   sample_collection_date >= mystartdate,
   sample_collection_date <= myenddate
  )

#Use as a check for size of data set later:
nrow(meta)

#Replace startdate with
 startdate <- as.Date(mystartdate)
#Set enddate
 maxdate<-as.Date(myenddate)

#Skipping the startdate line in duotang.Rmd, continue entering commands until "## 2. LOAD epidemiological data (PHAC)"

####################
#Adjust code first in plot_selection_estimator.R and save
####################
#Hard code refdate to the middle of the time period (avoids low counts of some variants setting time to a different point)
#and allows a consistent time at which the allele frequencies are measured across provinces (here the middate
 refdate<-135

#which corresponds to April 15, 2022 for t=0

####################
#Adapting code from selection.Rmd do the following (only):
####################
source("scripts/plot_selection_estimator.R")

#define starting parameters (p,s) for making the selection estimates
startpar <- list(p=c(0.1, 0.05, 0.01), s=c(0.05, 0.05, 0.01))

#filter samples with variants of interest
setAll=getAllStrictoLineages(meta)
sublineages_BA45 <- c(getStrictoSubLineages("BA.5*",meta), getStrictoSubLineages("BA.4*",meta) )

sublineages_BA2 <- getStrictoSubLineages("BA.2*",meta)
```

```
#The rest in this period are primarily BA.1:
sublineages_BA1<-setAll[!setAll %in% c(sublineages_BA2, sublineages_BA45)] #the rest

sublineages_BA11  <- getStrictoSubLineages("BA.1.1*",meta)
sublineages_BA1 <- sublineages_BA1[!sublineages_BA1 %in% c(sublineages_BA11)] #BA.1 Other

reference <- sublineages_BA1
#define the mutants
mutants = list(sublineages_BA11, sublineages_BA2, sublineages_BA45)
#this list defines the mutants we are looking at for plotting. The order of mutants listed will be
respected on the respective plots.
#!!!last element is always the reference.
mutantNames = list("BA.1.1*",  "BA.2*", "BA.4/5*", "BA.1*")
col <- c("BA.1.1"="#CC5500","BA.2"=pal["Omicron BA.2"],"BA.4/5"=pal["Omicron BA.5"],"BA.1"=-
pal["Omicron BA.1"]) #define

#check that all lineages are present once:
nrow(setAll)-nrow(reference)-nrow(Reduce(union,mutants))

####################
#Switch to code from plot_selection_estimator.R and enter one region at a time then the following
code
####################
region<-"Ontario"
region<-"Alberta"
region<-"British Columbia"
region<-"Quebec"
method<-'BFGS'
collapseMutants<-T
sample_collection_date<-max(meta$sample_collection_date)

#Enter
 est <- .make.estimator(region, startdate, reference, mutants)
 toplot <- est$toplot
 toplot$tot <- apply(toplot[which(!is.element(names(toplot), c('time', 'date')))], 1, sum)

 fit <- .fit.model(est, startpar, method=method)
 fit

 # To visualize fits, enter code for "sub.plot.selection.estimate <- function(region,maxdate=NA){" in
selection.Rmd then enter:
```

sub.plot.selection.estimate(region, maxdate)

# Check that all sequences are accounted for
nrow(meta %>% filter(province == region))-(sum(toplot$n1)+sum(toplot$n2)+sum(toplot$n3)+sum(toplot$n4))

ONTARIO

p1	p2	p3	s1	s2	s3
0.118989176	0.872081681	0.006027493	0.037031381	0.094360512	0.168612635

In[∘]:=
```
ONML1 = ImportString[
   "0.118989176 0.872081681 0.006027493 0.037031381 0.094360512 0.168612635",
   "Table"][[1]]
```

Out[∘]=
```
{0.118989, 0.872082, 0.00602749, 0.0370314, 0.0943605, 0.168613}
```

ALBERTA

p1	p2	p3	s1	s2	s3
0.136079546	0.845212232	0.003240639	0.026057559	0.071230512	0.151728890

In[∘]:=
```
ABML1 = ImportString[
   "0.136079546 0.845212232 0.003240639 0.026057559 0.071230512 0.151728890 ",
   "Table"][[1]]
```

Out[∘]=
```
{0.13608, 0.845212, 0.00324064, 0.0260576, 0.0712305, 0.151729}
```

BRITISH COLUMBIA

p1	p2	p3	s1	s2	s3
0.062429749	0.929763426	0.002509135	0.033689743	0.095625477	0.177918711

In[∘]:=
```
BCML1 = ImportString[
   "0.062429749 0.929763426 0.002509135 0.033689743 0.095625477 0.177918711 ",
   "Table"][[1]]
```

Out[∘]=
```
{0.0624297, 0.929763, 0.00250914, 0.0336897, 0.0956255, 0.177919}
```

QUEBEC

p1	p2	p3	s1	s2	s3
0.130481532	0.848160875	0.002500278	0.028521735	0.080077145	0.167335046

In[∘]:=
```
QCML1 = ImportString[
   "0.130481532 0.848160875 0.002500278 0.028521735 0.080077145 0.167335046 ",
   "Table"][[1]]
```

Out[∘]=
```
{0.130482, 0.848161, 0.00250028, 0.0285217, 0.0800771, 0.167335}
```

duotang: Lineage evolution from September 1 2022 - May 31 2023

```
###################
#Ran duotang in RStudio, adjusting dates and lineages of interest to estimate the selection coefficients and lineage frequencies for different periods of time.
#Version : 22 September 2023
#Obtained from https://github.com/CoVaRR-NET/duotang
###################

###################
#Notes to run duotang.Rmd in RStudio :
#Set directories and dates needed to run duotang.Rmd
###################
setwd ("~/Dropbox/COVID19/CoVaRRNet_Pillar6_Notebook")
params <- list ()
params$datadir <-"data_needed/"
params$datestamp <-"2023-09-02"

#Desired date range
mystartdate <-“2022-09-01”
myenddate <-“2023-05-31”
params$datestamp <- myenddate

#Enter duotang.Rmd through “meta$month” then adjust meta to trim data after the desired date

 meta <- meta %>% filter(
   sample_collection_date >= mystartdate,
   sample_collection_date <= myenddate
  )

 #Use as a check for size of data set later:
 nrow(meta)

 #Replace startdate with
  startdate <- as.Date(mystartdate)
 #Set enddate
  maxdate<-as.Date(myenddate)

 #Skipping the startdate line in duotang.Rmd, continue entering commands until “## 2. LOAD epidemiological data (PHAC)”

###################
#Adjust code first in plot_selection_estimator.R and save
```

```
####################
#Hard code refdate to the middle of the time period (avoids low counts of some variants setting time to
a different point)
#and allows a consistent time at which the allele frequencies are measured across provinces (here the
middate
 refdate<-135

#which corresponds to Jan 14, 2023 for t=0

####################
#Adapting code from selection.Rmd do the following (only):
####################
source("scripts/plot_selection_estimator.R")

#define starting parameters (p,s) for making the selection estimates
startpar <- list(p=c(0.1, 0.05, 0.01), s=c(0.05, 0.05, 0.01))

#filter samples with variants of interest
setAll=getAllStrictoLineages(meta)
sublineages_XBB <- getStrictoSubLineages("XBB*",meta) #XBB variants

sublineages_XBB15  <- getStrictoSubLineages("XBB.1.5*",meta) #XBB1.5 variants
sublineages_XBBOther <- sublineages_XBB[!sublineages_XBB %in% c(sublineages_XBB15)] #XBB Other

sublineages_BQ <- getStrictoSubLineages("BQ*",meta)

#The rest in this period are primarily BA.5 & BA.4:
sublineages_Other<-setAll[!setAll %in% c(sublineages_BQ, sublineages_XBB15, sublineages_XB-
BOther)] #the rest

reference <- sublineages_Other
#define the mutants
mutants = list(sublineages_XBB15, sublineages_XBBOther, sublineages_BQ)
#this list defines the mutants we are looking at for plotting. The order of mutants listed will be
respected on the respective plots.
#!!!last element is always the reference.
mutantNames = list("XBB.1.5*",  "XBB.Other*", "BQ*", "BA.5*") #
col <- c("XBB.1.5*"="#00eeff","XBB.Other"=pal["XBB"],"BQ" = pal["Omicron BQ"]) #define custom color
for XBB here because otherwise it will be black cuz recombinant.

#check that all lineages are present once:
```

```
nrow(setAll)-nrow(reference)-nrow(Reduce(union,mutants))

###################
#Switch to code from plot_selection_estimator.R and enter one region at a time then the following code
###################
region<-“Ontario”
region<-“Alberta”
region<-“British Columbia”
region<-“Quebec”
method<-‘BFGS’
collapseMutants<-T
sample_collection_date<-max(meta$sample_collection_date)

#Enter
 est <- .make.estimator(region, startdate, reference, mutants)
 toplot <- est$toplot
 toplot$tot <- apply(toplot[which(!is.element(names(toplot), c(‘time’, ‘date’)))], 1, sum)

 fit <- .fit.model(est, startpar, method=method)
 fit

 # To visualize fits, enter code for “sub.plot.selection.estimate <- function(region,maxdate=NA){“ in selection.Rmd then enter:
 sub.plot.selection.estimate(region, maxdate)

 # Check that all sequences are accounted for
 nrow(meta %>% filter(province == region))-(sum(toplot$n1)+sum(toplot$n2)+sum(toplot$n3)+sum(toplot$n4))
```

```
ONTARIO
     p1       p2       p3       s1       s2       s3
0.129498579 0.007625208 0.634857867 0.074372906 0.091417510 0.032032796

$confint
     2.5 %    97.5 %
p1 0.124208973 0.134788185
p2 0.006643921 0.008606496
p3 0.627474991 0.642240743
s1 0.073088066 0.075657747
s2 0.089509689 0.093325331
```

s3 0.031330440 0.032735153

```
In[∘]:= ONML2 = ImportString[
    "0.129498579 0.007625208 0.634857867 0.074372906 0.091417510 0.032032796",
    "Table"][[1]]
Out[∘]= {0.129499, 0.00762521, 0.634858, 0.0743729, 0.0914175, 0.0320328}
```

ALBERTA
$fit
p1 p2 p3 s1 s2 s3
0.09118951 0.01216224 0.73510214 0.07422397 0.08730765 0.03323601

$confint
2.5 % 97.5 %
p1 0.082596403 0.09978262
p2 0.009555296 0.01476919
p3 0.722050410 0.74815387
s1 0.071921601 0.07652633
s2 0.084149996 0.09046531
s3 0.032005843 0.03446618

```
In[∘]:= ABML2 = ImportString[
    "0.09118951 0.01216224 0.73510214 0.07422397 0.08730765 0.03323601 ",
    "Table"][[1]]
Out[∘]= {0.0911895, 0.0121622, 0.735102, 0.074224, 0.0873077, 0.033236}
```

QUEBEC
$fit
p1 p2 p3 s1 s2 s3
0.170114876 0.005888324 0.579924511 0.063673467 0.082866917 0.023798106

$confint
2.5 % 97.5 %
p1 0.160389348 0.179840403
p2 0.004627387 0.007149262
p3 0.567772865 0.592076158
s1 0.062035976 0.065310958
s2 0.080205256 0.085528578
s3 0.022916588 0.024679624

In[∘]:=
```
QCML2 = ImportString[
   "0.170114876 0.005888324 0.579924511 0.063673467 0.082866917 0.023798106",
   "Table"][[1]]
```

Out[∘]=
```
{0.170115, 0.00588832, 0.579925, 0.0636735, 0.0828669, 0.0237981}
```

BRITISH COLUMBIA

p1	p2	p3	s1	s2	s3
0.085192517	0.005515947	0.681707601	0.075028809	0.093109572	0.028918971

$confint

	2.5 %	97.5 %
p1	0.077013733	0.093371301
p2	0.004167128	0.006864766
p3	0.668738202	0.694677000
s1	0.072886130	0.077171487
s2	0.089883560	0.096335584
s3	0.027855483	0.029982460

In[∘]:=
```
BCML2 = ImportString[
   "0.085192517 0.005515947 0.681707601 0.075028809 0.093109572 0.028918971 ",
   "Table"][[1]]
```

Out[∘]=
```
{0.0851925, 0.00551595, 0.681708, 0.0750288, 0.0931096, 0.028919}
```

Canadian age data

Canadian population size by 70+ and all ages on 1 July 2022 (from https://www150.statcan.gc.ca/n1/t-bl/csv/17100005-eng.zip):

GEO	Total	70+
Alberta	4543111	439109
British Columbia	5319324	726247
Canada	38929902	5022509
Manitoba	1409223	159565
New Brunswick	812061	125512
Newfoundland and Labrador	525972	83596
Northwest Territories	45605	2507
Nova Scotia	1019725	152435
Nunavut	40526	1076
Ontario	15109416	1918812
Prince Edward Island	170688	23926
Quebec	8695659	1250508
Saskatchewan	1194803	135543

Yukon 43789 3673

In[∘]:=
```
CanadaPop = ImportString["GEO	Total	70+
AB	4543111	439109
BC	5319324	726247
Canada	38929902	5022509
MB	1409223	159565
NB	812061	125512
NL	525972	83596
NT	45605	2507
NS	1019725	152435
NU	40526	1076
ON	15109416	1918812
PE	170688	23926
QC	8695659	1250508
SK	1194803	135543
YT	43789	3673", "Table"];
```

In[∘]:=
```
CanadaPop // MatrixForm
```

Out[∘]//MatrixForm=

GEO	Total	70+
AB	4 543 111	439 109
BC	5 319 324	726 247
Canada	38 929 902	5 022 509
MB	1 409 223	159 565
NB	812 061	125 512
NL	525 972	83 596
NT	45 605	2507
NS	1 019 725	152 435
NU	40 526	1076
ON	15 109 416	1 918 812
PE	170 688	23 926
QC	8 695 659	1 250 508
SK	1 194 803	135 543
YT	43 789	3673

BC - 70+ case counts and genomic frequencies (Dec 2021-May 2023)

http://www.bccdc.ca/Health-Info-Site/Documents/BCCDC_COVID19_Dashboard_Case_Details.csv

Processing BC Case Data [ENTER this or the next folder - uploading & processing data]

In[∘]:=
```
SetDirectory["/Users/otto/Dropbox/COVID19/BC_Data/CorrectingCaseCounts"];
```

In[∘]:=
```
startafter = "2021-11-30";
```

The two periods we'll consider:

In[∘]:=
```
mystartdate1 = "2021-12-01";
myenddate1 = "2022-08-31";
mystartdate2 = "2022-09-01";
myenddate2 = "2023-05-31";
```

In[∘]:=
```
curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

In[∘]:=
```
xaxis = {{curdate["2021-10-01"], "Oct"},
   {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
   {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
   {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
   {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
   {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
   {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
   {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
   {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
   {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
   {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

Uploading the data (see below to avoid this step):

In[∘]:=
```
Drop[Import["BCCDC_COVID19_Dashboard_Case_Details11May2023.csv"], 1];
temp = %[[200 000 ;; Length[%]]]; (*Limiting data*)
counts = Transpose[{temp[[All, 1]], temp[[All, 2]], temp[[All, 4]]}];
```

In[∘]:=
```
For[i = 1, i ≤ Length[counts], i++,
 counts[[i, 1]] =
  DateString[DateObject[counts[[i, 1]]], {"Year", "-", "Month", "-", "Day"}]
]
```

Out[∘]=
```
$Aborted
```

Making sure that we've included enough of the counts (next line should be before startafter date)

In[∘]:=
```
counts[[1]]
```

Out[∘]=
```
{2021-10-20, Northern, <10}
```

In[∘]:=
```
Last[Position[counts, startafter]][[1]]
```

Out[∘]=
```
18 932
```

In[∘]:=
```
counts = counts[[% + 1 ;; Length[counts]]];
```

In[∘]:=
```
dates = DeleteDuplicates[counts[[All, 1]]]
```

Out[∘]=
```
{2021-12-01, 2021-12-02, 2021-12-03, 2021-12-04, 2021-12-05, 2021-12-06, 2021-12-07,
 2021-12-08, 2021-12-09, 2021-12-10, 2021-12-11, 2021-12-12, 2021-12-13, 2021-12-14,
 2021-12-15, 2021-12-16, 2021-12-17, 2021-12-18, 2021-12-19, 2021-12-20, 2021-12-21,
```

```
2021-12-22, 2021-12-23, 2021-12-24, 2021-12-25, 2021-12-26, 2021-12-27,
2021-12-28, 2021-12-29, 2021-12-30, 2021-12-31, 2022-01-01, 2022-01-02,
2022-01-03, 2022-01-04, 2022-01-05, 2022-01-06, 2022-01-07, 2022-01-08,
2022-01-09, 2022-01-10, 2022-01-11, 2022-01-12, 2022-01-13, 2022-01-14,
2022-01-15, 2022-01-16, 2022-01-17, 2022-01-18, 2022-01-19, 2022-01-20,
2022-01-21, 2022-01-22, 2022-01-23, 2022-01-24, 2022-01-25, 2022-01-26,
2022-01-27, 2022-01-28, 2022-01-29, 2022-01-30, 2022-01-31, 2022-02-01,
2022-02-02, 2022-02-03, 2022-02-04, 2022-02-05, 2022-02-06, 2022-02-07,
2022-02-08, 2022-02-09, 2022-02-10, 2022-02-11, 2022-02-12, 2022-02-13,
2022-02-14, 2022-02-15, 2022-02-16, 2022-02-17, 2022-02-18, 2022-02-19,
2022-02-20, 2022-02-21, 2022-02-22, 2022-02-23, 2022-02-24, 2022-02-25,
2022-02-26, 2022-02-27, 2022-02-28, 2022-03-01, 2022-03-02, 2022-03-03,
2022-03-04, 2022-03-05, 2022-03-06, 2022-03-07, 2022-03-08, 2022-03-09,
2022-03-10, 2022-03-11, 2022-03-12, 2022-03-13, 2022-03-14, 2022-03-15,
2022-03-16, 2022-03-17, 2022-03-18, 2022-03-19, 2022-03-20, 2022-03-21,
2022-03-22, 2022-03-23, 2022-03-24, 2022-03-25, 2022-03-26, 2022-03-27,
2022-03-28, 2022-03-29, 2022-03-30, 2022-03-31, 2022-04-01, 2022-04-02,
2022-04-03, 2022-04-04, 2022-04-05, 2022-04-06, 2022-04-07, 2022-04-08,
2022-04-09, 2022-04-10, 2022-04-11, 2022-04-12, 2022-04-13, 2022-04-14,
2022-04-15, 2022-04-16, 2022-04-17, 2022-04-18, 2022-04-19, 2022-04-20,
2022-04-21, 2022-04-22, 2022-04-23, 2022-04-24, 2022-04-25, 2022-04-26,
2022-04-27, 2022-04-28, 2022-04-29, 2022-04-30, 2022-05-01, 2022-05-02,
2022-05-03, 2022-05-04, 2022-05-05, 2022-05-06, 2022-05-07, 2022-05-08,
2022-05-09, 2022-05-10, 2022-05-11, 2022-05-12, 2022-05-13, 2022-05-14,
2022-05-15, 2022-05-16, 2022-05-17, 2022-05-18, 2022-05-19, 2022-05-20,
2022-05-21, 2022-05-22, 2022-05-23, 2022-05-24, 2022-05-25, 2022-05-26,
2022-05-27, 2022-05-28, 2022-05-29, 2022-05-30, 2022-05-31, 2022-06-01,
2022-06-02, 2022-06-03, 2022-06-04, 2022-06-05, 2022-06-06, 2022-06-07,
2022-06-08, 2022-06-09, 2022-06-10, 2022-06-11, 2022-06-12, 2022-06-13,
2022-06-14, 2022-06-15, 2022-06-16, 2022-06-17, 2022-06-18, 2022-06-19,
2022-06-20, 2022-06-21, 2022-06-22, 2022-06-23, 2022-06-24, 2022-06-25,
2022-06-26, 2022-06-27, 2022-06-28, 2022-06-29, 2022-06-30, 2022-07-01,
2022-07-02, 2022-07-03, 2022-07-04, 2022-07-05, 2022-07-06, 2022-07-07,
2022-07-08, 2022-07-09, 2022-07-10, 2022-07-11, 2022-07-12, 2022-07-13,
2022-07-14, 2022-07-15, 2022-07-16, 2022-07-17, 2022-07-18, 2022-07-19,
2022-07-20, 2022-07-21, 2022-07-22, 2022-07-23, 2022-07-24, 2022-07-25,
2022-07-26, 2022-07-27, 2022-07-28, 2022-07-29, 2022-07-30, 2022-07-31,
2022-08-01, 2022-08-02, 2022-08-03, 2022-08-04, 2022-08-05, 2022-08-06,
2022-08-07, 2022-08-08, 2022-08-09, 2022-08-10, 2022-08-11, 2022-08-12,
2022-08-13, 2022-08-14, 2022-08-15, 2022-08-16, 2022-08-17, 2022-08-18,
2022-08-19, 2022-08-20, 2022-08-21, 2022-08-22, 2022-08-23, 2022-08-24,
2022-08-25, 2022-08-26, 2022-08-27, 2022-08-28, 2022-08-29, 2022-08-30,
2022-08-31, 2022-09-01, 2022-09-02, 2022-09-03, 2022-09-04, 2022-09-05,
2022-09-06, 2022-09-07, 2022-09-08, 2022-09-09, 2022-09-10, 2022-09-11,
```

```
2022-09-12, 2022-09-13, 2022-09-14, 2022-09-15, 2022-09-16, 2022-09-17,
2022-09-18, 2022-09-19, 2022-09-20, 2022-09-21, 2022-09-22, 2022-09-23,
2022-09-24, 2022-09-25, 2022-09-26, 2022-09-27, 2022-09-28, 2022-09-29,
2022-09-30, 2022-10-01, 2022-10-02, 2022-10-03, 2022-10-04, 2022-10-05,
2022-10-06, 2022-10-07, 2022-10-08, 2022-10-09, 2022-10-10, 2022-10-11,
2022-10-12, 2022-10-13, 2022-10-14, 2022-10-15, 2022-10-16, 2022-10-17,
2022-10-18, 2022-10-19, 2022-10-20, 2022-10-21, 2022-10-22, 2022-10-23,
2022-10-24, 2022-10-25, 2022-10-26, 2022-10-27, 2022-10-28, 2022-10-29,
2022-10-30, 2022-10-31, 2022-11-01, 2022-11-02, 2022-11-03, 2022-11-04,
2022-11-05, 2022-11-06, 2022-11-07, 2022-11-08, 2022-11-09, 2022-11-10,
2022-11-11, 2022-11-12, 2022-11-13, 2022-11-14, 2022-11-15, 2022-11-16,
2022-11-17, 2022-11-18, 2022-11-19, 2022-11-20, 2022-11-21, 2022-11-22,
2022-11-23, 2022-11-24, 2022-11-25, 2022-11-26, 2022-11-27, 2022-11-28,
2022-11-29, 2022-11-30, 2022-12-01, 2022-12-02, 2022-12-03, 2022-12-04,
2022-12-05, 2022-12-06, 2022-12-07, 2022-12-08, 2022-12-09, 2022-12-10,
2022-12-11, 2022-12-12, 2022-12-13, 2022-12-14, 2022-12-15, 2022-12-16,
2022-12-17, 2022-12-18, 2022-12-19, 2022-12-20, 2022-12-21, 2022-12-22,
2022-12-23, 2022-12-24, 2022-12-25, 2022-12-26, 2022-12-27, 2022-12-28,
2022-12-29, 2022-12-30, 2022-12-31, 2023-01-01, 2023-01-02, 2023-01-03,
2023-01-04, 2023-01-05, 2023-01-06, 2023-01-07, 2023-01-08, 2023-01-09,
2023-01-10, 2023-01-11, 2023-01-12, 2023-01-13, 2023-01-14, 2023-01-15,
2023-01-16, 2023-01-17, 2023-01-18, 2023-01-19, 2023-01-20, 2023-01-21,
2023-01-22, 2023-01-23, 2023-01-24, 2023-01-25, 2023-01-26, 2023-01-27,
2023-01-28, 2023-01-29, 2023-01-30, 2023-01-31, 2023-02-01, 2023-02-02,
2023-02-03, 2023-02-04, 2023-02-05, 2023-02-06, 2023-02-07, 2023-02-08,
2023-02-09, 2023-02-10, 2023-02-11, 2023-02-12, 2023-02-13, 2023-02-14,
2023-02-15, 2023-02-16, 2023-02-17, 2023-02-18, 2023-02-19, 2023-02-20,
2023-02-21, 2023-02-22, 2023-02-23, 2023-02-24, 2023-02-25, 2023-02-26,
2023-02-27, 2023-02-28, 2023-03-01, 2023-03-02, 2023-03-03, 2023-03-04,
2023-03-05, 2023-03-06, 2023-03-07, 2023-03-08, 2023-03-09, 2023-03-10,
2023-03-11, 2023-03-12, 2023-03-13, 2023-03-14, 2023-03-15, 2023-03-16,
2023-03-17, 2023-03-18, 2023-03-19, 2023-03-20, 2023-03-21, 2023-03-22,
2023-03-23, 2023-03-24, 2023-03-25, 2023-03-26, 2023-03-27, 2023-03-28,
2023-03-29, 2023-03-30, 2023-03-31, 2023-04-01, 2023-04-02, 2023-04-03,
2023-04-04, 2023-04-05, 2023-04-06, 2023-04-07, 2023-04-08, 2023-04-09,
2023-04-10, 2023-04-11, 2023-04-12, 2023-04-13, 2023-04-14, 2023-04-15}
```

Here we are going to combine the Health Authorities:

```
In[•]:= HAlist = {"Vancouver Coastal", "Vancouver Island",
    "Interior", "Fraser", "Northern", "Out of Canada"};
```

To adjust to the population size of the province:

```
In[∘]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "BC"][[1, 1]]]]
Out[∘]= {BC, 5 319 324, 726 247}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[∘]:= per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
        per100000A = 100 000 / popsize[[3]]
Out[∘]= 100 000 / 4 593 077
Out[∘]= 100 000 / 726 247
```

```
In[∘]:= Clear[datetabBC];
        datetabBC[i_] = {0, 0};
        For[i = 1, i ≤ Length[counts], i++,
         date = Position[dates, counts[[i, 1]]][[1, 1]];
         If[((counts[[i, 3]] == "70-79") ||
             (counts[[i, 3]] == "80-89") || (counts[[i, 3]] == "90+")),
          Part[datetabBC[date] = datetabBC[date] + {0, per100000A}],
          Part[datetabBC[date] = datetabBC[date] + {per100000Y, 0}]
         ]
        ]
```

Processing BC Case Data [ENTER this or the previous folder - pre-processed data]

```
In[∘]:= startafter = "2021-11-30";
```

The two periods we'll consider:

```
In[∘]:= mystartdate1 = "2021-12-01";
        myenddate1 = "2022-08-31";
        mystartdate2 = "2022-09-01";
        myenddate2 = "2023-05-31";
```

```
In[∘]:= curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

```
In[∘]:= xaxis = {{curdate["2021-10-01"], "Oct"},
    {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
    {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
    {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
    {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
    {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
    {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
    {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
    {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
    {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
    {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

To adjust to the population size of the province:

```
In[∘]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "BC"][[1, 1]]]]
Out[∘]= {BC, 5 319 324, 726 247}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[∘]:= per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
        per100000A = 100 000 / popsize[[3]]
```

Out[∘]= $\frac{100\,000}{4\,593\,077}$

Out[∘]= $\frac{100\,000}{726\,247}$

```
In[∘]:= tabdates = {"2021-12-01", "2021-12-02", "2021-12-03",
    "2021-12-04", "2021-12-05", "2021-12-06", "2021-12-07",
    "2021-12-08", "2021-12-09", "2021-12-10", "2021-12-11",
    "2021-12-12", "2021-12-13", "2021-12-14", "2021-12-15", "2021-12-16",
    "2021-12-17", "2021-12-18", "2021-12-19", "2021-12-20", "2021-12-21",
    "2021-12-22", "2021-12-23", "2021-12-24", "2021-12-25", "2021-12-26",
    "2021-12-27", "2021-12-28", "2021-12-29", "2021-12-30", "2021-12-31",
    "2022-01-01", "2022-01-02", "2022-01-03", "2022-01-04", "2022-01-05",
    "2022-01-06", "2022-01-07", "2022-01-08", "2022-01-09", "2022-01-10",
    "2022-01-11", "2022-01-12", "2022-01-13", "2022-01-14", "2022-01-15",
    "2022-01-16", "2022-01-17", "2022-01-18", "2022-01-19", "2022-01-20",
    "2022-01-21", "2022-01-22", "2022-01-23", "2022-01-24", "2022-01-25",
    "2022-01-26", "2022-01-27", "2022-01-28", "2022-01-29", "2022-01-30",
    "2022-01-31", "2022-02-01", "2022-02-02", "2022-02-03", "2022-02-04",
    "2022-02-05", "2022-02-06", "2022-02-07", "2022-02-08", "2022-02-09",
    "2022-02-10", "2022-02-11", "2022-02-12", "2022-02-13", "2022-02-14",
    "2022-02-15", "2022-02-16", "2022-02-17", "2022-02-18", "2022-02-19",
    "2022-02-20", "2022-02-21", "2022-02-22", "2022-02-23", "2022-02-24",
```

```
"2022-02-25", "2022-02-26", "2022-02-27", "2022-02-28", "2022-03-01",
"2022-03-02", "2022-03-03", "2022-03-04", "2022-03-05", "2022-03-06",
"2022-03-07", "2022-03-08", "2022-03-09", "2022-03-10", "2022-03-11",
"2022-03-12", "2022-03-13", "2022-03-14", "2022-03-15", "2022-03-16",
"2022-03-17", "2022-03-18", "2022-03-19", "2022-03-20", "2022-03-21",
"2022-03-22", "2022-03-23", "2022-03-24", "2022-03-25", "2022-03-26",
"2022-03-27", "2022-03-28", "2022-03-29", "2022-03-30", "2022-03-31",
"2022-04-01", "2022-04-02", "2022-04-03", "2022-04-04", "2022-04-05",
"2022-04-06", "2022-04-07", "2022-04-08", "2022-04-09", "2022-04-10",
"2022-04-11", "2022-04-12", "2022-04-13", "2022-04-14", "2022-04-15",
"2022-04-16", "2022-04-17", "2022-04-18", "2022-04-19", "2022-04-20",
"2022-04-21", "2022-04-22", "2022-04-23", "2022-04-24", "2022-04-25",
"2022-04-26", "2022-04-27", "2022-04-28", "2022-04-29", "2022-04-30",
"2022-05-01", "2022-05-02", "2022-05-03", "2022-05-04", "2022-05-05",
"2022-05-06", "2022-05-07", "2022-05-08", "2022-05-09", "2022-05-10",
"2022-05-11", "2022-05-12", "2022-05-13", "2022-05-14", "2022-05-15",
"2022-05-16", "2022-05-17", "2022-05-18", "2022-05-19", "2022-05-20",
"2022-05-21", "2022-05-22", "2022-05-23", "2022-05-24", "2022-05-25",
"2022-05-26", "2022-05-27", "2022-05-28", "2022-05-29", "2022-05-30",
"2022-05-31", "2022-06-01", "2022-06-02", "2022-06-03", "2022-06-04",
"2022-06-05", "2022-06-06", "2022-06-07", "2022-06-08", "2022-06-09",
"2022-06-10", "2022-06-11", "2022-06-12", "2022-06-13", "2022-06-14",
"2022-06-15", "2022-06-16", "2022-06-17", "2022-06-18", "2022-06-19",
"2022-06-20", "2022-06-21", "2022-06-22", "2022-06-23", "2022-06-24",
"2022-06-25", "2022-06-26", "2022-06-27", "2022-06-28", "2022-06-29",
"2022-06-30", "2022-07-01", "2022-07-02", "2022-07-03", "2022-07-04",
"2022-07-05", "2022-07-06", "2022-07-07", "2022-07-08", "2022-07-09",
"2022-07-10", "2022-07-11", "2022-07-12", "2022-07-13", "2022-07-14",
"2022-07-15", "2022-07-16", "2022-07-17", "2022-07-18", "2022-07-19",
"2022-07-20", "2022-07-21", "2022-07-22", "2022-07-23", "2022-07-24",
"2022-07-25", "2022-07-26", "2022-07-27", "2022-07-28", "2022-07-29",
"2022-07-30", "2022-07-31", "2022-08-01", "2022-08-02", "2022-08-03",
"2022-08-04", "2022-08-05", "2022-08-06", "2022-08-07", "2022-08-08",
"2022-08-09", "2022-08-10", "2022-08-11", "2022-08-12", "2022-08-13",
"2022-08-14", "2022-08-15", "2022-08-16", "2022-08-17", "2022-08-18",
"2022-08-19", "2022-08-20", "2022-08-21", "2022-08-22", "2022-08-23",
"2022-08-24", "2022-08-25", "2022-08-26", "2022-08-27", "2022-08-28",
"2022-08-29", "2022-08-30", "2022-08-31", "2022-09-01", "2022-09-02",
"2022-09-03", "2022-09-04", "2022-09-05", "2022-09-06", "2022-09-07",
"2022-09-08", "2022-09-09", "2022-09-10", "2022-09-11", "2022-09-12",
"2022-09-13", "2022-09-14", "2022-09-15", "2022-09-16", "2022-09-17",
"2022-09-18", "2022-09-19", "2022-09-20", "2022-09-21", "2022-09-22",
"2022-09-23", "2022-09-24", "2022-09-25", "2022-09-26", "2022-09-27",
"2022-09-28", "2022-09-29", "2022-09-30", "2022-10-01", "2022-10-02",
```

```
"2022-10-03", "2022-10-04", "2022-10-05", "2022-10-06", "2022-10-07",
"2022-10-08", "2022-10-09", "2022-10-10", "2022-10-11", "2022-10-12",
"2022-10-13", "2022-10-14", "2022-10-15", "2022-10-16", "2022-10-17",
"2022-10-18", "2022-10-19", "2022-10-20", "2022-10-21", "2022-10-22",
"2022-10-23", "2022-10-24", "2022-10-25", "2022-10-26", "2022-10-27",
"2022-10-28", "2022-10-29", "2022-10-30", "2022-10-31", "2022-11-01",
"2022-11-02", "2022-11-03", "2022-11-04", "2022-11-05", "2022-11-06",
"2022-11-07", "2022-11-08", "2022-11-09", "2022-11-10", "2022-11-11",
"2022-11-12", "2022-11-13", "2022-11-14", "2022-11-15", "2022-11-16",
"2022-11-17", "2022-11-18", "2022-11-19", "2022-11-20", "2022-11-21",
"2022-11-22", "2022-11-23", "2022-11-24", "2022-11-25", "2022-11-26",
"2022-11-27", "2022-11-28", "2022-11-29", "2022-11-30", "2022-12-01",
"2022-12-02", "2022-12-03", "2022-12-04", "2022-12-05", "2022-12-06",
"2022-12-07", "2022-12-08", "2022-12-09", "2022-12-10", "2022-12-11",
"2022-12-12", "2022-12-13", "2022-12-14", "2022-12-15", "2022-12-16",
"2022-12-17", "2022-12-18", "2022-12-19", "2022-12-20", "2022-12-21",
"2022-12-22", "2022-12-23", "2022-12-24", "2022-12-25", "2022-12-26",
"2022-12-27", "2022-12-28", "2022-12-29", "2022-12-30", "2022-12-31",
"2023-01-01", "2023-01-02", "2023-01-03", "2023-01-04", "2023-01-05",
"2023-01-06", "2023-01-07", "2023-01-08", "2023-01-09", "2023-01-10",
"2023-01-11", "2023-01-12", "2023-01-13", "2023-01-14", "2023-01-15",
"2023-01-16", "2023-01-17", "2023-01-18", "2023-01-19", "2023-01-20",
"2023-01-21", "2023-01-22", "2023-01-23", "2023-01-24", "2023-01-25",
"2023-01-26", "2023-01-27", "2023-01-28", "2023-01-29", "2023-01-30",
"2023-01-31", "2023-02-01", "2023-02-02", "2023-02-03", "2023-02-04",
"2023-02-05", "2023-02-06", "2023-02-07", "2023-02-08", "2023-02-09",
"2023-02-10", "2023-02-11", "2023-02-12", "2023-02-13", "2023-02-14",
"2023-02-15", "2023-02-16", "2023-02-17", "2023-02-18", "2023-02-19",
"2023-02-20", "2023-02-21", "2023-02-22", "2023-02-23", "2023-02-24",
"2023-02-25", "2023-02-26", "2023-02-27", "2023-02-28", "2023-03-01",
"2023-03-02", "2023-03-03", "2023-03-04", "2023-03-05", "2023-03-06",
"2023-03-07", "2023-03-08", "2023-03-09", "2023-03-10", "2023-03-11",
"2023-03-12", "2023-03-13", "2023-03-14", "2023-03-15", "2023-03-16",
"2023-03-17", "2023-03-18", "2023-03-19", "2023-03-20", "2023-03-21",
"2023-03-22", "2023-03-23", "2023-03-24", "2023-03-25", "2023-03-26",
"2023-03-27", "2023-03-28", "2023-03-29", "2023-03-30", "2023-03-31",
"2023-04-01", "2023-04-02", "2023-04-03", "2023-04-04", "2023-04-05",
"2023-04-06", "2023-04-07", "2023-04-08", "2023-04-09", "2023-04-10",
"2023-04-11", "2023-04-12", "2023-04-13", "2023-04-14", "2023-04-15"};
```

tab = Table[datetabBC[i], {i, 1, Length[dates]}] // N

```
In[•]:= tab = {{7.5548483075724615`, 2.891578209617389`}, {8.44749609031157`,
   2.3408014077855053`}, {7.641935896132375`, 1.3769420045797092`},
  {6.683972421973331`, 2.3408014077855053`}, {6.161446890613852`,
```

```
 2.6161898087014475`}, {5.921956022074091`, 1.2392478041217383`},
{7.924970558952093`, 1.51463620503768`}, {7.162954159052853`,
 1.652330405495651`}, {9.492547153030529`, 3.02927241007536`},
{9.623178535870398`, 2.6161898087014475`}, {8.012058147512006`,
 2.0654130068695635`}, {7.445988821872571`, 2.6161898087014475`},
{8.817618341691201`, 2.0654130068695635`}, {12.649472238327379`,
 2.2031072073275344`}, {15.958800603604077`, 3.166966610533331`},
{17.02562356346301`, 2.4784956082434766`}, {19.790654500240255`,
 3.30466081099l302`}, {18.506112568981536`, 3.166966610533331`},
{18.157762214741883`, 3.166966610533331`}, {26.27867984795378`,
 4.130826013739127`}, {33.09328365276698`, 4.54390861511304`},
{43.39139099997671`, 5.6454622187768075`}, {53.319376095806795`,
 7.297792624272458`}, {55.10467166128502`, 5.920850619692749`},
{44.13163550273597`, 9.08781723022608`}, {37.92664481784216`,
 5.507768018318837`}, {40.32155350323977`, 6.609321621982604`},
{65.29391952279485`, 11.841701239385499`}, {84.25724193171592`,
 16.936386656330424`}, {80.2729847550999`, 18.588717061826074`},
{85.06280212589512`, 24.50956768151882`}, {66.16479540839399`,
 24.09648508014491`}, {47.440963868012666`, 20.791824269153608`},
{54.49505854136562`, 24.09648508014491`}, {68.2548975338319`,
 24.371873481060852`}, {68.12426615099203`, 27.53884009159418`},
{62.98609842595715`, 30.981195103043454`}, {52.644447284467475`,
 25.88650968609853`}, {47.72399853083238`, 32.49583130808114`},
{42.23748045155786`, 29.741947298921716`}, {39.42890572050066`,
 29.053476296631864`}, {52.40495641592771`, 40.34440073418548`},
{50.597888953309514`, 34.974326916324614`}, {43.347847205696745`,
 36.07588051998838`}, {46.52654418813358`, 40.75748333555939`},
{35.22692957248485`, 32.77121970899708`}, {29.065482681871`, 30.2927241007536`},
{33.528721595566545`, 33.45969071128693`},
{47.65868283941245`, 41.30826013739127`},
{41.693183023058396`, 33.45969071128693`},
{45.895159171074205`, 33.32199651082896`},
{40.952938520299135`, 38.69207032868982`},
{30.698374967369368`, 24.647261881976792`},
{26.12627656797393`, 31.118889303501426`},
{25.23362878523482`, 28.77808789571592`}, {37.27348790364281`,
 34.5612443149507`}, {35.074526292505006`, 33.04660810991302`},
{37.3823473893427`, 39.655929731895625`}, {31.612794647248457`,
 31.945054506249253`}, {24.558699973895497`, 27.53884009159418`},
{19.96482967736008`, 25.060344483350704`}, {19.9430577802201`,
 29.466558898005776`}, {33.15859934418692`, 34.285855914034755`},
{29.39206113897067`, 28.36500529434201`}, {28.738904224771325`,
 24.784956082434764`}, {24.428068591055627`, 28.089616893426065`},
{21.532406271438514`, 27.814228492510125`}, {15.828169220764206`,
```

```
 21.20490687052752`}, {15.458046969384576`, 23.683402478770997`},
{23.09998286551695`, 25.88650968609853`}, {24.972366019555082`,
 26.850369089304326`}, {17.918271346202122`, 27.951922692968093`},
{16.350694752123687`, 24.50956768151882`}, {11.952771529848073`,
 22.031072073275347`}, {10.25456355292976 9`, 18.31332866091013`},
{11.843912044148182`, 18.726411262284042`}, {11.952771529848073`,
 23.8210966792289 7`}, {12.018087221268008`, 18.31332866091013`},
{12.192262398387834`, 23.132625676939114`}, {9.862669404410159`,
 18.17563446045216`}, {7.859654867532158`, 15.00866784991883`},
{6.44448155343357`, 13.080949043507237`}, {6.204990684893809`,
 14.733279449002888`}, {8.948249724531072`, 17.349469257704335`},
{10.450510627189573`, 20.241047467321724`}, {8.752302650271266`,
 16.110221453582596`}, {6.967007084793049`, 15.972527253124627`},
{5.3123429021546995`, 12.667866442133324`}, {4.963992547915047`,
 11.704007038927527`}, {6.44448155343357`, 10.46475923480579`},
{8.055601941791963`, 13.494031644881149`}, {6.531569141993483`,
 10.740147635721732`}, {5.573605667834439`, 11.015536036637673`},
{5.3123429021546995`, 12.530172241675352`}, {4.68095788509533`,
 10.051676633431876`}, {3.461731645256546`, 11.566312838469557`},
{4.223748045155785`, 11.566312838469557`}, {4.572098399395438`,
 10.60245343526376`}, {5.290571005014721`, 11.566312838469557`},
{4.746273576515264`, 10.327065034347818`}, {3.8318538966361766`,
 8.123957827020284`}, {2.7214871424972844`, 5.370073817860866`},
{2.6997152453573063`, 8.95012302976811`}, {3.8318538966361766`,
 8.399346227936226`}, {4.027800970895981`, 8.399346227936226`},
{3.766538205216242`, 7.160098423814487`}, {3.8536257937761547`,
 5.507768018318837`}, {3.3528721595566546`, 6.333933221066662`},
{2.852118525337154`, 7.022404223356516`}, {2.2425054054177624`,
 6.19623902060869 1`}, {3.9842571766160244`, 8.123957827020284`},
{4.093116662315916`, 9.363205631142021`}, {4.180204250875829`,
 10.327065034347818`}, {3.4181878509765893`, 8.812428829310138`},
{3.2004688795768064`, 9.500899831599993`}, {3.1133812910168936`,
 9.913982432973906`}, {2.3513648911176537`, 8.674734628852168`},
{3.8318538966361766`, 9.776288232515935`}, {4.41969511941559`,
 12.805560642591296`}, {4.158432353735851`, 9.225511430684051`},
{4.8333611650751775`, 9.638594032057965`}, {3.1786969824368283`,
 13.76942004579709`}, {2.677943348217328`, 10.189370833889848`},
{3.548819233816459`, 11.428618638011585`}, {3.5270473366764805`,
 14.595585248544916`}, {3.004521805317002`, 10.740147635721732`},
{4.136660456595872`, 12.392478041217382`}, {3.9842571766160244`,
 11.15323023709564 3`}, {3.875397690916133`, 11.015536036637673`},
{3.004521805317002`, 13.907114246255063`}, {3.4181878509765893`,
 14.595585248544916`}, {3.1351531881568717`, 14.595585248544916`},
{4.615642193675394`, 17.762551859078247`}, {3.9842571766160244`,
```

```
 17.07408085678839`}, {4.441467016555569`, 17.349469257704335`},
{3.9189414851960898`, 14.182502647171004`}, {3.244012673856763`,
 15.559444651750713`}, {3.091609393876915`, 13.494031644881149`},
{2.8085747310571976`, 18.72641126228404`}, {5.987271713494026`,
 18.588717061826074`}, {4.942220650775068`, 19.827964865947813`},
{4.898676856495112`, 21.34260107098549`}, {4.506782707975503`,
 21.61798947190143`}, {3.1133812910168936`, 13.218643243965207`},
{2.8956623196171107`, 15.834833052666655`}, {4.637414090815373`,
 23.270319877397085`}, {5.5518337706944605`, 22.719543075565202`},
{4.332607530855677`, 21.34260107098549`}, {4.724501679375286`,
 20.103353266863753`}, {3.439959748116568`, 14.595585248544916`},
{2.852118525337154`, 16.79869245587245`}, {2.982749908177024`,
 14.320196847628974`}, {3.352872159556546`, 19.27718806411593`},
{3.5923630280964156`, 19.96565906640578`}, {4.071344765175938`,
 17.487163458162307`}, {3.9624852794760463`, 19.139493863657957`},
{3.5052754395365024`, 17.211775057246363`}, {2.743259039637263`,
 12.667866442133324`}, {1.6764360797783273`, 14.044808446713033`},
{3.22240776716785`, 14.182502647171004`}, {3.4835035423965244`,
 17.211775057246363`}, {2.9609780110370454`, 15.697138852208683`},
{3.1786969824368283`, 16.79869245587245`}, {2.52554006823748`,
 14.595585248544916`}, {2.373136788257632`, 10.877841836179702`},
{1.7417517711982622`, 11.841701239385499`}, {2.547311965377458`,
 13.494031644881149`}, {2.852118525337154`, 11.979395439843469`},
{2.5908557596574147`, 12.392478041217382`}, {2.198961611137806`,
 11.841701239385499`}, {2.068330228297936`, 12.25478384075941`},
{1.8288393597581751`, 13.080949043507237`}, {1.5893484912184141`,
 12.667866442133324`}, {1.7852955654782185`, 8.537040428394196`},
{2.4602243768175454`, 13.63172584533912`}, {2.6343995539373712`,
 9.363205631142021`}, {1.9376988454580666`, 11.153230237095643`},
{2.0247864340179795`, 9.638594032057965`}, {1.502260902658501`,
 6.7470158224405745`}, {1.502260902658501`, 7.5731810251884`},
{1.415173314098588`, 9.776288232515935`}, {1.5675765940784359`,
 11.015536036637673`}, {1.5893484912184141`, 9.225511430684051`},
{1.5458046969384576`, 7.435486824730429`}, {1.6764360797783273`,
 6.471627421524633`}, {1.1321386512788703`, 5.6454622187768075`},
{1.2845419312587183`, 6.471627421524633`}, {1.3934014169586098`,
 4.956991216486953`}, {1.719979874058284`, 7.022404223356516`},
{1.2192262398387834`, 6.196239020608691`}, {1.2192262398387834`,
 5.783156419234778`}, {1.502260902658501`, 6.884710022898545`},
{1.088594856998914`, 3.02927241007536`}, {1.2192262398387834`,
 3.9931318132811566`}, {1.719979874058284`, 6.7470158224405745`},
{1.5675765940784359`, 5.6454622187768075`}, {1.023279165578979`,
 5.094685416944924`}, {1.023279165578979`, 3.304660810991302`},
{1.502260902658501`, 4.6816028155710105`}, {0.8055601941791962`,
```

2.891578209617389`}, {0.8708758855991311`, 4.130826013739127`},
{1.0450510627189573`, 4.54390861511304`}, {1.3280857255386749`,
6.05854482015072`}, {1.0668229598589356`, 8.123957827020284`},
{1.3934014169586098`, 4.956991216486953`}, {0.9579634741590441`,
5.370073817860866`}, {0.8926477827391093`, 4.54390861511304`},
{1.1974543426988051`, 4.54390861511304`}, {1.6111203883583924`,
4.406214414655069`}, {1.6111203883583924`, 5.920850619692749`},
{1.26277003411874`, 6.7470158224405745`}, {1.3063138283986966`,
6.884710022898545`}, {1.502260902658501`, 9.225511430684051`},
{1.26277003411874`, 6.196239020608691`}, {1.5675765940784359`,
5.6454622187768075`}, {1.8288393597581751`, 9.776288232515935`},
{2.0030145368780015`, 8.261652027478254`}, {1.719979874058284`,
8.95012302976811`}, {1.7417517711982622`, 9.913982432973906`},
{1.7417517711982622`, 7.8485694261043415`}, {1.1539105484188485`,
9.225511430684051`}, {1.5893484912184141`, 8.261652027478254`},
{1.719979874058284`, 9.363205631142021`}, {1.7417517711982622`,
7.8485694261043415`}, {2.2207335082777844`, 9.500899831599993`},
{2.3949086853976103`, 11.290924437553615`}, {1.415173314098588`,
8.261652027478254`}, {1.6764360797783273`, 8.674734628852168`},
{1.6328922854983707`, 7.297792624272458`}, {1.9812426397380232`,
5.6454622187768075`}, {1.7635236683382403`, 9.776288232515935`},
{1.8288393597581751`, 10.60245343526376`}, {1.5458046969384576`,
7.022404223356516`}, {1.8070674626181968`, 8.399346227936226`},
{0.9797353712990224`, 5.094685416944924`}, {1.5240327997984793`,
7.297792624272458`}, {2.068330228297936`, 8.399346227936226`},
{2.177189713997828`, 11.979395439843469`}, {1.8723831540381317`,
8.674734628852168`}, {1.5458046969384576`, 9.225511430684051`},
{1.5240327997984793`, 8.399346227936226`}, {1.1321386512788703`,
6.471627421524633`}, {1.1103667541388922`, 8.537040428394196`},
{1.2192262398387834`, 7.8485694261043415`}, {2.046558331157958`,
10.60245343526376`}, {1.9376988454580666`, 9.08781723022608`},
{1.502260902658501`, 7.986263626562313`}, {1.2845419312587183`,
7.710875225646371`}, {1.0450510627189573`, 7.8485694261043415`},
{1.1756824455588268`, 10.740147635721732`}, {0.8491039884591528`,
7.5731810251884`}, {1.3280857255386749`, 10.877841836179702`},
{1.6764360797783273`, 9.638594032057965`}, {1.26277003411874`,
13.218643243965207`}, {1.3063138283986966`, 7.710875225646371`},
{1.1974543426988051`, 5.920850619692749`}, {0.8491039884591528`,
7.022404223356516`}, {1.3934014169586098`, 5.783156419234778`},
{1.3280857255386749`, 10.051676633431876`}, {1.1756824455588268`,
6.609321621982604`}, {1.088594856998914`, 8.537040428394196`},
{1.023279165578979`, 9.776288232515935`}, {0.7402445027592613`,
5.6454622187768075`}, {0.7402445027592613`, 6.05854482015072`},
{0.8273320913191745`, 10.189370833889848`}, {0.8926477827391093`,

```
 5.920850619692749`}, {1.2409981369787617`, 6.884710022898545`},
{1.3498576226786532`, 6.884710022898545`}, {0.9144196798790876`,
 4.54390861511304`}, {0.9797353712990224`, 5.232379617402895`},
{0.8926477827391093`, 4.819297016028982`}, {0.9361915770190659`,
 6.33393322106662`}, {1.1539105484188485`, 7.160098423814487`},
{1.1321386512788703`, 7.8485694261043415`}, {0.8273320913191745`,
 6.471627421524633`}, {0.8708758855991311`, 6.196239020608691`},
{0.8273320913191745`, 4.130826013739127`}, {0.9361915770190659`,
 4.130826013739127`}, {0.6967007084793049`, 4.819297016028982`},
{0.6967007084793049`, 7.297792624272458`}, {1.0668229598589356`,
 6.05854482015072`}, {1.0450510627189573`, 8.537040428394196`},
{0.6749288113393266`, 6.05854482015072`}, {0.6749288113393266`,
 5.370073817860866`}, {0.9361915770190659`, 5.783156419234778`},
{1.023279165578979`, 7.297792624272458`}, {1.023279165578979`,
 6.196239020608691`}, {0.8926477827391093`, 6.609321621982604`},
{0.9144196798790876`, 8.399346227936226`}, {0.9579634741590441`,
 6.471627421524633`}, {0.5660693256394351`, 6.7470158224405745`},
{0.6967007084793049`, 5.094685416944924`}, {0.8273320913191745`,
 8.537040428394196`}, {0.6749288113393266`, 5.507768018318837`},
{0.9144196798790876`, 6.196239020608691`}, {1.1539105484188485`,
 11.428618638011585`}, {0.8491039884591528`, 6.7470158224405745`},
{0.8491039884591528`, 6.196239020608691`}, {0.7402445027592613`,
 8.812428829310138`}, {1.0450510627189573`, 8.123957827020284`},
{1.1539105484188485`, 9.638594032057965`}, {0.9579634741590441`,
 7.5731810251884`}, {0.8708758855991311`, 10.189370833889848`},
{0.5225255313594787`, 7.160098423814487`}, {0.6967007084793049`,
 6.471627421524633`}, {0.7184726056192831`, 7.160098423814487`},
{0.8926477827391093`, 7.435486824730429`}, {0.8055601941791962`,
 11.841701239385499`}, {0.6096131199193917`, 12.25478384075941`},
{0.7837882970392179`, 7.986263626562313`}, {0.5660693256394351`,
 8.95012302976811`}, {0.5225255313594787`, 9.08781723022608`},
{0.7837882970392179`, 8.123957827020284`}, {0.7620163998992396`,
 7.5731810251884`}, {0.6096131199193917`, 8.674734628852168`},
{0.8055601941791962`, 12.11708964030144`}, {0.5660693256394351`,
 7.022404223356516`}, {0.5878412227794134`, 6.196239020608691`},
{0.6096131199193917`, 6.884710022898545`}, {0.5225255313594787`,
 6.609321621982604`}, {0.41366604565958726`, 6.05854482015072`},
{0.8708758855991311`, 7.160098423814487`}, {0.63138501705937`,
 7.710875225646371`}, {0.6967007084793049`, 6.609321621982604`},
{0.6749288113393266`, 6.05854482015072`}, {0.4789817370795221`,
 6.05854482015072`}, {0.5660693256394351`, 5.094685416944924`},
{0.5007536342195004`, 6.471627421524633`}, {0.39189414851960896`,
 5.232379617402895`}, {0.6749288113393266`, 6.884710022898545`},
{0.544297428499457`, 7.160098423814487`}, {0.5007536342195004`,
```

```
 8.123957827020284`}, {0.63138501705937`, 3.7177434123652144`},
{0.39189414851960896`, 4.819297016028982`}, {0.23949086853976104`,
 6.05854482015072`}, {0.6531569141993483`, 4.54390861511304`},
{0.6749288113393266`, 7.022404223356516`}, {0.21771897139978277`,
 3.4423550114492727`}, {0.6967007084793049`, 3.9931318132811566`},
{0.5660693256394351`, 4.6816028155710105`}, {0.544297428499457`,
 5.370073817860866`}, {0.6531569141993483`, 5.507768018318837`},
{0.5007536342195004`, 7.022404223356516`}, {0.6967007084793049`,
 6.333933221066662`}, {0.4572098399395438`, 7.710875225646371`},
{0.41366604565958726`, 6.05854482015072`}, {0.5660693256394351`,
 4.6816028155710105`}, {0.43543794279956555`, 5.920850619692749`},
{0.39189414851960896`, 5.094685416944924`}, {0.7620163998992396`,
 7.710875225646371`}, {0.5660693256394351`, 6.333933221066662`},
{0.30480655995969586`, 6.884710022898545`}, {0.8708758855991311`,
 7.022404223356516`}, {0.544297428499457`, 5.507768018318837`},
{0.5007536342195004`, 7.160098423814487`}, {0.8273320913191745`,
 9.363205631142021`}, {0.5878412227794134`, 7.5731810251884`},
{0.544297428499457`, 6.471627421524633`}, {0.9797353712990224`,
 8.674734628852168`}, {0.5878412227794134`, 8.261652027478254`},
{0.6096131199193917`, 6.609321621982604`}, {0.4572098399395438`,
 6.609321621982604`}, {0.6531569141993483`, 6.884710022898545`},
{0.8055601941791962`, 8.399346227936226`}, {0.6531569141993483`,
 6.7470158224405745`}, {0.6096131199193917`, 7.022404223356516`},
{0.6967007084793049`, 6.05854482015072`}, {0.34835035423965244`,
 5.920850619692749`}, {0.43543794279956555`, 5.6454622187768075`},
{0.6749288113393266`, 8.537040428394196`}, {0.5660693256394351`,
 9.500899831599993`}, {0.6749288113393266`, 10.740147635721732`},
{0.9579634741590441`, 7.5731810251884`}, {1.0015072684390007`,
 7.8485694261043415`}, {0.63138501705937`, 9.776288232515935`},
{0.5660693256394351`, 8.399346227936226`}, {0.9361915770190659`,
 7.710875225646371`}, {0.6096131199193917`, 8.399346227936226`},
{0.6531569141993483`, 7.8485694261043415`}, {0.5007536342195004`,
 8.537040428394196`}, {0.7184726056192831`, 7.8485694261043415`},
{0.5007536342195004`, 6.7470158224405745`}, {0.5007536342195004`,
 6.471627421524633`}, {0.7620163998992396`, 6.884710022898545`},
{0.8273320913191745`, 6.05854482015072`}, {0.6531569141993483`,
 6.884710022898545`}, {0.6096131199193917`, 9.500899831599993`},
{0.41366604565958726`, 4.406214414655069`}, {0.7620163998992396`,
 8.537040428394196`}, {0.5660693256394351`, 7.710875225646371`},
{0.5878412227794134`, 7.986263626562313`}, {0.5007536342195004`,
 10.740147635721732`}, {0.63138501705937`, 7.5731810251884`},
{1.0015072684390007`, 11.566312838469557`}, {0.9579634741590441`,
 10.740147635721732`}, {0.6096131199193917`, 8.399346227936226`},
{0.4572098399395438`, 9.08781723022608`}, {0.6967007084793049`,
```

```
 7.8485694261043415`}, {0.6096131199193917`, 10.327065034347818`},
{0.6967007084793049`, 11.290924437553615`}, {0.6531569141993483`,
 10.051676633431876`}, {0.8055601941791962`, 8.399346227936226`},
{0.544297428499457`, 5.920850619692749`}, {0.63138501705937`,
 4.130826013739127`}, {0.4572098399395438`, 10.46475923480579`},
{0.6531569141993483`, 8.95012302976811`}, {0.63138501705937`,
 6.884710022898545`}, {0.39189414851960896`, 7.435486824730429`},
{0.6531569141993483`, 8.399346227936226`}, {0.6096131199193917`,
 5.232379617402895`}, {0.4789817370795221`, 7.710875225646371`},
{0.5007536342195004`, 8.95012302976811`}, {0.39189414851960896`,
 4.406214414655069`}, {0.30480655995969586`, 5.094685416944924`},
{0.3701222513796307`, 3.5800492119072436`}, {0.6531569141993483`,
 3.166966610533331`}, {0.39189414851960896`, 3.7177434123652144`},
{0.544297428499457`, 2.891578209617389`}, {0.30480655995969586`,
 1.7900246059536218`}, {0.5007536342195004`, 2.4784956082434766`},
{0.7402445027592613`, 3.304660810991302`}, {0.5660693256394351`,
 1.3769420045797092`}, {0.7402445027592613`, 2.891578209617389`},
{0.32657845709967415`, 1.9277188064115927`}, {0.30480655995969586`,
 1.3769420045797092`}, {0.63138501705937`, 2.3408014077855053`},
{0.32657845709967415`, 3.7177434123652144`}, {0.43543794279956555`,
 4.6816028155710105`}, {0.4789817370795221`, 2.6161898087014475`},
{0.30480655995969586`, 3.02927241007536`}, {0.26126276567973933`,
 3.02927241007536`}, {0.30480655995969586`, 2.7538840091594183`},
{0.43543794279956555`, 3.7177434123652144`}, {0.63138501705937`,
 2.891578209617389`}, {0.5878412227794134`, 3.02927241007536`},
{0.26126276567973933`, 5.370073817860866`}, {0.5225255313594787`,
 4.130826013739127`}, {0.4572098399395438`, 4.819297016028982`},
{0.5007536342195004`, 3.5800492119072436`}, {0.39189414851960896`,
 5.6454622187768075`}, {0.19594707425980448`, 3.7177434123652144`},
{0.4789817370795221`, 4.819297016028982`}, {0.6531569141993483`,
 6.47162742152463`}, {0.28303466281971756`, 2.7538840091594183`},
{0.41366604565958726`, 3.5800492119072436`}, {0.34835035423965244`,
 4.406214414655069`}, {0.39189414851960896`, 4.6816028155710105`},
{0.6531569141993483`, 7.5731810251884`}, {0.41366604565958726`,
 4.268520214197098`}, {0.41366604565958726`, 5.232379617402895`},
{0.39189414851960896`, 5.920850619692749`}, {0.41366604565958726`,
 4.406214414655069`}, {0.39189414851960896`, 5.783156419234778`},
{0.41366604565958726`, 3.7177434123652144`}, {0.4789817370795221`,
 5.6454622187768075`}, {0.30480655995969586`, 4.268520214197098`},
{0.5878412227794134`, 4.819297016028982`}, {0.30480655995969586`,
 2.6161898087014475`}, {0.34835035423965244`, 2.891578209617389`},
{0.5007536342195004`, 4.130826013739127`}, {0.32657845709967415`,
 4.6816028155710105`}, {0.6531569141993483`, 5.6454622187768075`},
{0.39189414851960896`, 4.819297016028982`}, {0.3701222513796307`,
```

```
 3.8554376128231853`}, {0.4572098399395438`, 3.7177434123652144`},
{0.3701222513796307`, 5.232379617402895`}, {0.32657845709967415`,
 5.370073817860866`}, {0.544297428499457`, 3.5800492119072436`},
{0.6749288113393266`, 5.6454622187768075`}, {0.5007536342195004`,
 5.6454622187768075`}, {0.4789817370795221`, 4.268520214197098`},
{0.41366604565958726`, 5.232379617402895`}, {0.39189414851960896`,
 4.956991216486953`}, {0.43543794279956555`, 6.196239020608691`},
{0.5007536342195004`, 4.406214414655069`}, {0.41366604565958726`,
 7.297792624272458`}, {0.5007536342195004`, 5.370073817860866`},
{0.3701222513796307`, 7.022404223356516`}, {0.4572098399395438`,
 4.819297016028982`}, {0.43543794279956555`, 7.8485694261043415`},
{0.41366604565958726`, 7.022404223356516`}, {0.5878412227794134`,
 6.471627421524633`}, {0.5225255313594787`, 7.5731810251884`},
{0.39189414851960896`, 5.094685416944924`}, {0.4572098399395438`,
 5.370073817860866`}, {0.43543794279956555`, 5.370073817860866`},
{0.5660693256394351`, 4.819297016028982`}, {0.4572098399395438`,
 6.196239020608691`}, {0.43543794279956555`, 5.6454622187768075`},
{0.6096131199193917`, 5.920850619692749`}, {0.7184726056192831`,
 5.6454622187768075`}, {0.43543794279956555`, 6.609321621982604`},
{0.6531569141993483`, 5.783156419234778`}, {0.4572098399395438`,
 4.956991216486953`}, {0.5225255313594787`, 5.370073817860866`},
{1.0015072684390007`, 7.022404223356516`}, {0.5878412227794134`,
 5.920850619692749`}, {0.43543794279956555`, 6.196239020608691`},
{0.63138501705937`, 5.6454622187768075`}, {0.6967007084793049`,
 7.297792624272458`}, {0.6096131199193917`, 5.6454622187768075`}};
```

Enter the following to avoid upload:

```
In[•]:= dates = tabdates;
```

```
In[•]:= For[i = 1, i <= Length[dates], i++, datetabBC[i] = tab[[i]]]
```

Overall Analysis by 70+

Cubic Spline fit (https://mathematica.stackexchange.com/questions/33206/implementation-of-smoothing-splines-function)

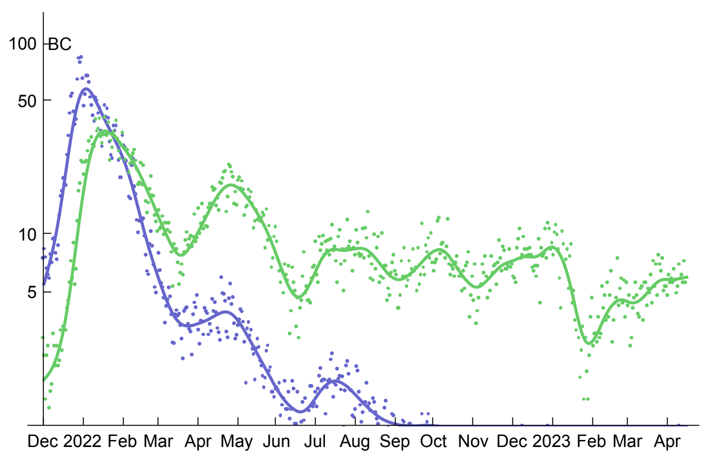
```
In[•]:= CubicSplSmooth[data_, lambda_] := Module[{M, Knots, X, Dsq, a}, M = Length@data;
   Knots = Flatten@{1, 1, 1, Range@M, M, M, M};
   X = Table[Evaluate@N@BSplineBasis[{3, Knots}, n, t], {t, 1, M}, {n, 0, M + 1}];
   Dsq = Differences[X, 2];
   a = LinearSolve[Transpose[X].X + lambda * Transpose[Dsq].Dsq,
     Transpose[X].data, Method → "Multifrontal"];
   Return[
    X.
     a]];
```

Cubic spline fit to all dates (Y is younger, A is 70+), using lambda = 4 to provide a good fit but not overfit to the data:

```
In[∘]:= modYBC = CubicSplSmooth[Table[{i, Log[Max[1, datetabBC[i][[1]]]]},
          {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
        modABC = CubicSplSmooth[Table[{i, Log[Max[1, datetabBC[i][[2]]]]},
          {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
```

```
In[∘]:= plotBC = Show[
          ListPlot[Table[{i, Log[datetabBC[i][[1]]]}, {i, 1, Length[dates]}],
           PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.4, 0.8]}],
          ListLinePlot[modYBC, PlotStyle → RGBColor[0.4, 0.4, 0.8], PlotRange → All],
          ListLinePlot[modABC, PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
          ListPlot[Table[{i, Log[datetabBC[i][[2]]]}, {i, 1, Length[dates]}],
           PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
          Graphics[Text["BC", {5, 4.6}, {-1, 0}]],
          PlotRange → {Automatic, {0, 5}}, AxesOrigin → {1, 0},
          Ticks → {xaxis, {{Log[5], "5"}, {Log[10], "10"}, {Log[50], "50"}, {Log[100], "100"},
             {Log[500], "500"}, {Log[1000], "1000"}, {Log[5000], "5000"}}}
         ]
```

Out[∘]=

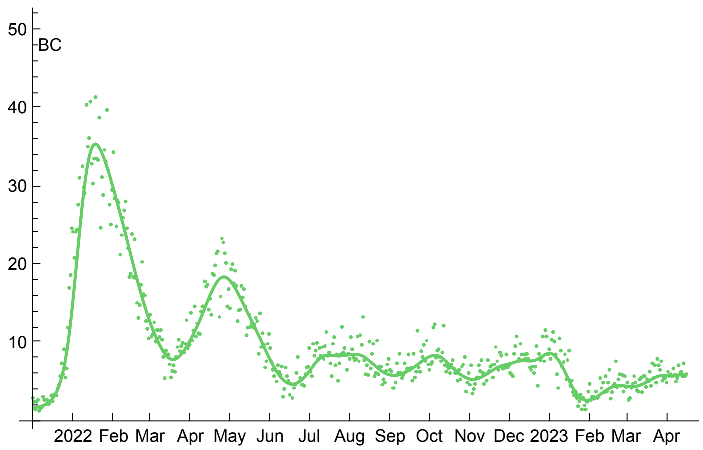


The above illustrates the major drop off in reported cases among those <70:

70+ on a non-log plot

In[∘]:=
```
Show[
 ListLinePlot[Table[{modABC[[i, 1]], Exp[modABC[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabBC[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["BC", {5, 48}, {-1, 0}]],
 PlotRange → {Automatic, {0, 50}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

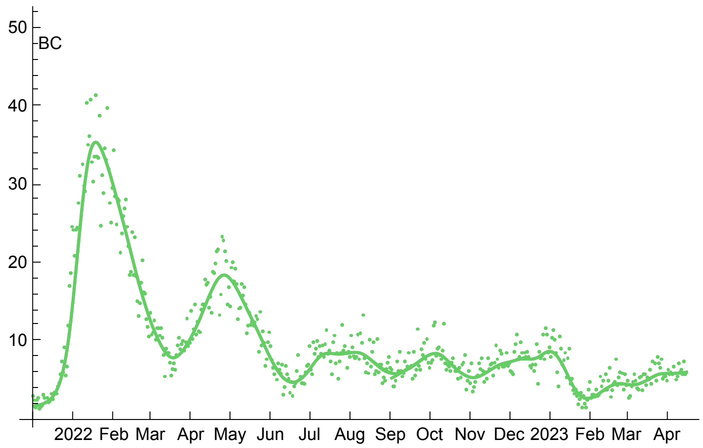
Out[∘]=



In[∘]:=
```
modABC1 = CubicSplSmooth[Table[{i, Log[Max[1, datetabBC[i][[2]]]]},
    {i, 1, Position[dates, myenddate1][[1, 1]]}], 10^lambda /. lambda → 3];
modABC2 = CubicSplSmooth[Table[{i, Log[Max[1, datetabBC[i][[2]]]]},
    {i, Position[dates, mystartdate2][[1, 1]], Length[dates]}],
   10^lambda /. lambda → 3];
```

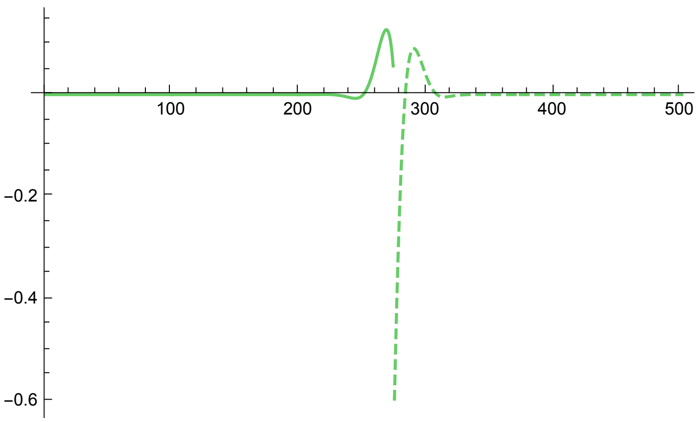
```
In[∘]:= Show[
  ListLinePlot[Table[{modABC[[i, 1]], Exp[modABC[[i, 2]]]}, {i, 1, Length[dates]}],
   PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
  ListLinePlot[Table[{modABC1[[i, 1]], Exp[modABC1[[i, 2]]]},
    {i, 1, Length[modABC1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
  ListLinePlot[Table[{modABC2[[i, 1]], Exp[modABC2[[i, 2]]]},
    {i, 1, Length[modABC2]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
  ListPlot[Table[{i, datetabBC[i][[2]]}, {i, 1, Length[dates]}],
   PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
  Graphics[Text["BC", {5, 48}, {-1, 0}]],
  PlotRange → {Automatic, {0, 50}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
 ]
```

Out[∘]=



The fit is very similar, with a slight discrepancy in numbers at the joint, so we just use the overall spline fit for the analysis:

In[ ]:=
```
Show[
 ListLinePlot[Table[{modABC1[[i, 1]], Exp[modABC1[[i, 2]]] - Exp[modABC[[i, 2]]]},
   {i, 1, Length[modABC1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modABC2[[i, 1]], Exp[modABC2[[i, 2]]] - Exp[modABC[[
      i + Position[dates, myenddate1][[1, 1]], 2]]]}, {i, 1, Length[modABC2]}],
  PlotStyle → {Dashed, RGBColor[0.4, 0.8, 0.4]}, PlotRange → All],
 PlotRange → All
]
```

Out[ ]=


Merging with Omicron subvariant frequencies

In[ ]:=
```
graphheight = 42;
```

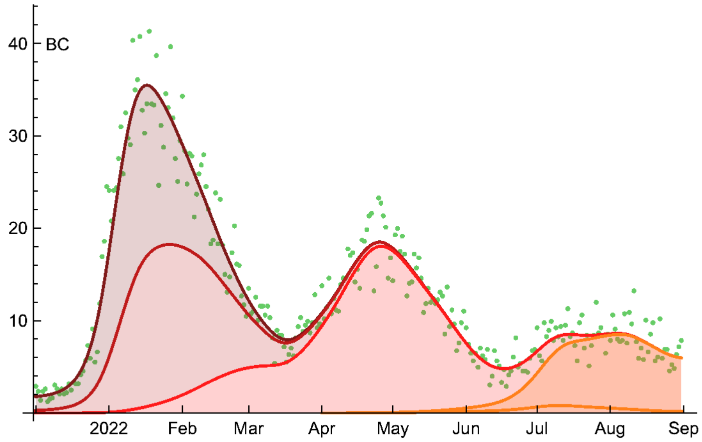
In[ ]:=
```
colourBA1 = RGBColor[0.5, 0.1, 0.1];
colourBA11 = RGBColor[0.75, 0.1, 0.1];
colourBA2 = RGBColor[1, 0.1, 0.1];
colourBA5 = RGBColor[1, 0.5, 0.1];
colourBA45 = colourBA5;
colourBA4 = colourBA5;
colourBQ = RGBColor[1, 0.75, 0.5];
colourXBB15 = RGBColor[0.2, 0.5, 0.8];
colourXBBO = RGBColor[0.5, 0.7, 0.8];

{colourBA1, colourBA11, colourBA2, colourBA45, colourBQ, colourXBB15, colourXBBO}
```

Out[ ]=


From Omicron_VariantsGISAID.nb (selection on BA.1, BA.1.1 and BA.2 calculated from 16 May analysis [uses “2023-12-1” for t=1]; selection on BA.4&BA.5 from 24 June analysis where the initial frequency is for “2022-04-1” at t=1):

```
In[◦]:= BCML = {pBA11 → 0.15389609946813929`, pBA2 → 0.00023336006270172828`,
    sBA11 → 0.03289439606417615`, sBA2 → 0.10253826895207825`,
    pBA4 → 0.00015574461261878253`, pBA5 → 0.000047964983160236866`,
    sBA4 → 0.08125393761607246`, sBA5 → 0.11284956139460468`};

In[◦]:= endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modABC[[i, 1]],
     1 * Exp[modABC[[i, 2]]] /. t -> modABC[[i, 1]] /. BCML}, {i, 1, endtime}];
tabBA11 = Table[{modABC[[i, 1]],
     (pBA2 Exp[sBA2 t] + pBA11 Exp[sBA11 t]) / ((1 - pBA11 - pBA2) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t]) * Exp[modABC[[i, 2]]] /.
       t -> modABC[[i, 1]] /. BCML}, {i, 1, endtime}];
tabBA2 = Table[{modABC[[i, 1]],
     (pBA2 Exp[sBA2 t]) / ((1 - pBA11 - pBA2) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t]) * Exp[modABC[[i, 2]]] /.
       t -> modABC[[i, 1]] /. BCML}, {i, 1, endtime}];
tabBA4 = Table[{modABC[[i, 1]], (pBA4 Exp[sBA4 t]) / ((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
        Exp[modABC[[i, 2]]] /. t -> modABC[[i, 1]] - curdate["2022-03-31"] /.
      BCML}, {i, curdate["2022-04-1"], endtime}];
tabBA5 = Table[{modABC[[i, 1]], (pBA5 Exp[sBA5 t] + pBA4 Exp[sBA4 t]) / ((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
        Exp[modABC[[i, 2]]] /. t -> modABC[[i, 1]] - curdate["2022-03-31"] /.
      BCML}, {i, curdate["2022-04-1"], endtime}];

Show[
 ListLinePlot[Table[{modABC[[i, 1]], Exp[modABC[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabBC[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA4, tabBA5}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
      Filling → {1 → {2}, 2 → {3}, 3 → Axis, 4 → Axis, 5 → {4}}],
 Graphics[Text["BC", {5, 0.95 graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[•]=



From duotang/VirusSeq (see code above): mutantNames = list("BA.1.1*", "BA.2*", "BA.4/5*", "BA.1*")

In[•]:=
```
BCML1
```

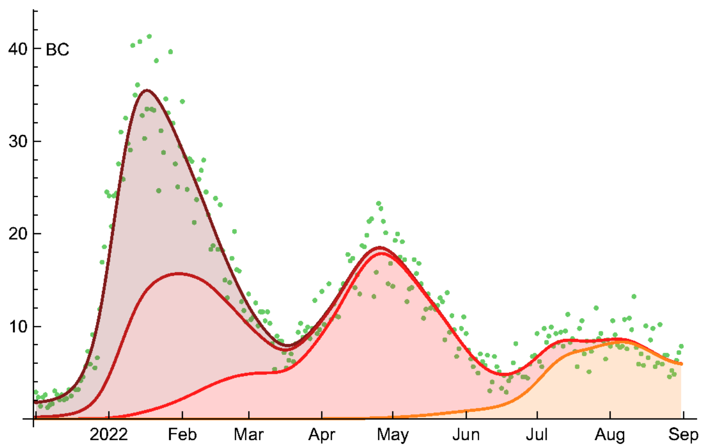
Out[•]=
```
{0.0624297, 0.929763, 0.00250914, 0.0336897, 0.0956255, 0.177919}
```

In[•]:=
```
dataname = BCML1;
refdate = "2022-04-15"; (* t=0 date used in duotang code above *)
BCML = {pBA11 → dataname[[1]], pBA2 → dataname[[2]], pBA45 → dataname[[3]],
   sBA11 → dataname[[4]], sBA2 → dataname[[5]], sBA45 → dataname[[6]]}
```

Out[•]=
```
{pBA11 → 0.0624297, pBA2 → 0.929763, pBA45 → 0.00250914,
 sBA11 → 0.0336897, sBA2 → 0.0956255, sBA45 → 0.177919}
```

In[•]:=
```
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modABC[[i, 1]],
    1 * Exp[modABC[[i, 2]]] /. t -> modABC[[i, 1]] /. BCML}, {i, 1, endtime}];
tabBA11 = Table[{modABC[[i, 1]],
    (pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modABC[[i, 2]]] /.
     t -> modABC[[i, 1]] - curdate[refdate] /. BCML}, {i, 1, endtime}];
tabBA2 = Table[{modABC[[i, 1]],
    (pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modABC[[i, 2]]] /.
     t -> modABC[[i, 1]] - curdate[refdate] /. BCML}, {i, 1, endtime}];
tabBA45 = Table[{modABC[[i, 1]],
    (pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modABC[[i, 2]]] /.
     t -> modABC[[i, 1]] - curdate[refdate] /. BCML}, {i, 1, endtime}];

Show[
 ListLinePlot[Table[{modABC[[i, 1]], Exp[modABC[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabBC[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
     Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis}],
 Graphics[Text["BC", {5, 0.95 graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[∘]=



From duotang/VirusSeq (see code above): mutantNames = list("XBB.1.5*", "XBB.Other*", "BQ*", "BA.5*")

In[∘]:=
```
BCML2
```

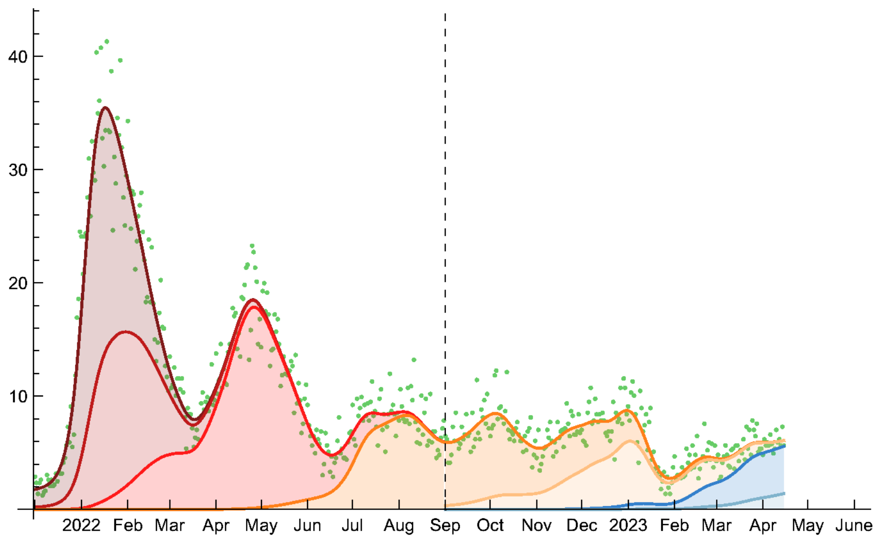
Out[∘]=
```
{0.0851925, 0.00551595, 0.681708, 0.0750288, 0.0931096, 0.028919}
```

In[∘]:=
```
dataname = BCML2;
refdate = "2023-01-14"; (* t=0 date used in duotang code above *)
BCML = {pXBB15 → dataname[[1]], pXBBO → dataname[[2]], pBQ → dataname[[3]],
  sXBB15 → dataname[[4]], sXBBO → dataname[[5]], sBQ → dataname[[6]]}
```

Out[∘]=
```
{pXBB15 → 0.0851925, pXBBO → 0.00551595, pBQ → 0.681708,
 sXBB15 → 0.0750288, sXBBO → 0.0931096, sBQ → 0.028919}
```

```
In[•]:= starttime = Position[dates, "2022-09-01"][[1, 1]];
endtime = Length[dates];

tabBA5 = Table[{modABC[[i, 1]], 1 * Exp[modABC[[i, 2]]] /. t -> modABC[[i, 1]] /. BCML},
   {i, starttime, endtime}];
tabBQ = Table[{modABC[[i, 1]],
     (pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
     ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
       Exp[modABC[[i, 2]]] /.
      t -> modABC[[i, 1]] - curdate[refdate] /. BCML}, {i, starttime, endtime}];
tabXBB15 = Table[{modABC[[i, 1]],
     (pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
     ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
       Exp[modABC[[i, 2]]] /.
      t -> modABC[[i, 1]] - curdate[refdate] /. BCML}, {i, starttime, endtime}];
tabXBBO = Table[{modABC[[i, 1]],
     (pXBBO Exp[sXBBO t]) /
     ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
       Exp[modABC[[i, 2]]] /.
      t -> modABC[[i, 1]] - curdate[refdate] /. BCML}, {i, starttime, endtime}];

Show[
 ListLinePlot[Table[{modABC[[i, 1]], Exp[modABC[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabBC[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45, tabBA5, tabBQ, tabXBB15, tabXBBO},
  PlotStyle → {colourBA1, colourBA11, colourBA2, colourBA45,
    colourBA45, colourBQ, colourXBB15, colourXBBO}, PlotRange → All,
      Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis,
    5 → {6}, 6 → {7}, 7 → {8}, 8 → Axis}],
 (*Graphics[Text["BC",{5,48},{-1,0}]],*)
 ListPlot[{{starttime, 0}, {starttime, 100}},
  Joined → True, PlotStyle → {Black, Thickness[0.002], Dashed}],
 PlotRange → {{1, curdate["2023-06-01"]}, {0, graphheight}},
 Ticks → {xaxis, Automatic},
 AspectRatio → 0.6,
 ImageSize → 450
]
```

Out[∘]=



QC - 70+ case counts and genomic frequencies (Dec 2021-May 2023)

https://www.inspq.qc.ca/covid-19/donnees/age-sexe/evolution-cas (1.1c - Carte thermique -> three dots -> “Télécharger les données en format CSV”.)

Processing QC Case Data [ENTER this or the next folder - uploading & processing data]

In[∘]:=
```
SetDirectory["/Users/otto/Dropbox/COVID19/BC_Data/CorrectingCaseCounts"];
```

In[∘]:=
```
SetDirectory["/Users/otto/Downloads"];
```

In[∘]:=
```
startafter = "2021-11-30";
```

The two periods we’ll consider:

In[∘]:=
```
mystartdate1 = "2021-12-01";
myenddate1 = "2022-08-31";
mystartdate2 = "2022-09-01";
myenddate2 = "2023-05-31";
```

In[∘]:=
```
curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

```
xaxis = {{curdate["2021-10-01"], "Oct"},
   {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
   {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
   {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
   {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
   {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
   {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
   {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
   {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
   {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
   {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

Uploading the data (see below to avoid this step):

```
In[•]:= temp0 = Import["graph_1-1.c_page_age_et_sexe_evol_des_cas22Sep2023.csv"];
temp0[[1]]
temp = Drop[temp0, 1];
counts = Table[{DateString[
     DateObject[temp[[i, 1]], DateFormat → {"Year", "-", "Month", "-", "Day"}]],
    temp[[i, 2]], temp[[i, 3]]}, {i, 1, Length[temp]}];
Clear[temp0, temp]

Out[•]= {"Date de résultat du test", y (y), y (value), Annotations 1}

In[•]:= counts = Sort[counts];

In[•]:= Last[Position[counts, startafter]][[1]]

Out[•]= 6780

In[•]:= counts = counts[[% + 1 ;; Length[counts]]];

In[•]:= counts[[1 ;; 10]]

Out[•]= {{2021-12-01, 0-9 ans, 366}, {2021-12-01, 10-19 ans, 224},
  {2021-12-01, 20-29 ans, 109}, {2021-12-01, 30-39 ans, 199},
  {2021-12-01, 40-49 ans, 162}, {2021-12-01, 50-59 ans, 101},
  {2021-12-01, 60-69 ans, 66}, {2021-12-01, 70-79 ans, 48},
  {2021-12-01, 80-89 ans, 13}, {2021-12-01, 90 ans et plus, 2}}
```

Trim off data from Quebec after end date (this data stream keeps going):

```
In[•]:= Last[Position[counts, myenddate2]][[1]]

Out[•]= 5470

In[•]:= counts = counts[[1 ;; %]];

In[•]:= dates = DeleteDuplicates[counts[[All, 1]]]

Out[•]= {2021-12-01, 2021-12-02, 2021-12-03, 2021-12-04, 2021-12-05, 2021-12-06, 2021-12-07,
  2021-12-08, 2021-12-09, 2021-12-10, 2021-12-11, 2021-12-12, 2021-12-13, 2021-12-14,
  2021-12-15, 2021-12-16, 2021-12-17, 2021-12-18, 2021-12-19, 2021-12-20, 2021-12-21,
```

2021-12-22, 2021-12-23, 2021-12-24, 2021-12-25, 2021-12-26, 2021-12-27, 2021-12-28,
2021-12-29, 2021-12-30, 2021-12-31, 2022-01-01, 2022-01-02, 2022-01-03,
2022-01-04, 2022-01-05, 2022-01-06, 2022-01-07, 2022-01-08, 2022-01-09,
2022-01-10, 2022-01-11, 2022-01-12, 2022-01-13, 2022-01-14, 2022-01-15,
2022-01-16, 2022-01-17, 2022-01-18, 2022-01-19, 2022-01-20, 2022-01-21,
2022-01-22, 2022-01-23, 2022-01-24, 2022-01-25, 2022-01-26, 2022-01-27,
2022-01-28, 2022-01-29, 2022-01-30, 2022-01-31, 2022-02-01, 2022-02-02,
2022-02-03, 2022-02-04, 2022-02-05, 2022-02-06, 2022-02-07, 2022-02-08,
2022-02-09, 2022-02-10, 2022-02-11, 2022-02-12, 2022-02-13, 2022-02-14,
2022-02-15, 2022-02-16, 2022-02-17, 2022-02-18, 2022-02-19, 2022-02-20,
2022-02-21, 2022-02-22, 2022-02-23, 2022-02-24, 2022-02-25, 2022-02-26,
2022-02-27, 2022-02-28, 2022-03-01, 2022-03-02, 2022-03-03, 2022-03-04,
2022-03-05, 2022-03-06, 2022-03-07, 2022-03-08, 2022-03-09, 2022-03-10,
2022-03-11, 2022-03-12, 2022-03-13, 2022-03-14, 2022-03-15, 2022-03-16,
2022-03-17, 2022-03-18, 2022-03-19, 2022-03-20, 2022-03-21, 2022-03-22,
2022-03-23, 2022-03-24, 2022-03-25, 2022-03-26, 2022-03-27, 2022-03-28,
2022-03-29, 2022-03-30, 2022-03-31, 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04,
2022-04-05, 2022-04-06, 2022-04-07, 2022-04-08, 2022-04-09, 2022-04-10,
2022-04-11, 2022-04-12, 2022-04-13, 2022-04-14, 2022-04-15, 2022-04-16,
2022-04-17, 2022-04-18, 2022-04-19, 2022-04-20, 2022-04-21, 2022-04-22,
2022-04-23, 2022-04-24, 2022-04-25, 2022-04-26, 2022-04-27, 2022-04-28,
2022-04-29, 2022-04-30, 2022-05-01, 2022-05-02, 2022-05-03, 2022-05-04,
2022-05-05, 2022-05-06, 2022-05-07, 2022-05-08, 2022-05-09, 2022-05-10,
2022-05-11, 2022-05-12, 2022-05-13, 2022-05-14, 2022-05-15, 2022-05-16,
2022-05-17, 2022-05-18, 2022-05-19, 2022-05-20, 2022-05-21, 2022-05-22,
2022-05-23, 2022-05-24, 2022-05-25, 2022-05-26, 2022-05-27, 2022-05-28,
2022-05-29, 2022-05-30, 2022-05-31, 2022-06-01, 2022-06-02, 2022-06-03,
2022-06-04, 2022-06-05, 2022-06-06, 2022-06-07, 2022-06-08, 2022-06-09,
2022-06-10, 2022-06-11, 2022-06-12, 2022-06-13, 2022-06-14, 2022-06-15,
2022-06-16, 2022-06-17, 2022-06-18, 2022-06-19, 2022-06-20, 2022-06-21,
2022-06-22, 2022-06-23, 2022-06-24, 2022-06-25, 2022-06-26, 2022-06-27,
2022-06-28, 2022-06-29, 2022-06-30, 2022-07-01, 2022-07-02, 2022-07-03, 2022-07-04,
2022-07-05, 2022-07-06, 2022-07-07, 2022-07-08, 2022-07-09, 2022-07-10,
2022-07-11, 2022-07-12, 2022-07-13, 2022-07-14, 2022-07-15, 2022-07-16,
2022-07-17, 2022-07-18, 2022-07-19, 2022-07-20, 2022-07-21, 2022-07-22,
2022-07-23, 2022-07-24, 2022-07-25, 2022-07-26, 2022-07-27, 2022-07-28,
2022-07-29, 2022-07-30, 2022-07-31, 2022-08-01, 2022-08-02, 2022-08-03,
2022-08-04, 2022-08-05, 2022-08-06, 2022-08-07, 2022-08-08, 2022-08-09,
2022-08-10, 2022-08-11, 2022-08-12, 2022-08-13, 2022-08-14, 2022-08-15,
2022-08-16, 2022-08-17, 2022-08-18, 2022-08-19, 2022-08-20, 2022-08-21,
2022-08-22, 2022-08-23, 2022-08-24, 2022-08-25, 2022-08-26, 2022-08-27,
2022-08-28, 2022-08-29, 2022-08-30, 2022-08-31, 2022-09-01, 2022-09-02,
2022-09-03, 2022-09-04, 2022-09-05, 2022-09-06, 2022-09-07, 2022-09-08,
2022-09-09, 2022-09-10, 2022-09-11, 2022-09-12, 2022-09-13, 2022-09-14,

```
2022-09-15, 2022-09-16, 2022-09-17, 2022-09-18, 2022-09-19, 2022-09-20,
2022-09-21, 2022-09-22, 2022-09-23, 2022-09-24, 2022-09-25, 2022-09-26, 2022-09-27,
2022-09-28, 2022-09-29, 2022-09-30, 2022-10-01, 2022-10-02, 2022-10-03,
2022-10-04, 2022-10-05, 2022-10-06, 2022-10-07, 2022-10-08, 2022-10-09,
2022-10-10, 2022-10-11, 2022-10-12, 2022-10-13, 2022-10-14, 2022-10-15,
2022-10-16, 2022-10-17, 2022-10-18, 2022-10-19, 2022-10-20, 2022-10-21,
2022-10-22, 2022-10-23, 2022-10-24, 2022-10-25, 2022-10-26, 2022-10-27,
2022-10-28, 2022-10-29, 2022-10-30, 2022-10-31, 2022-11-01, 2022-11-02,
2022-11-03, 2022-11-04, 2022-11-05, 2022-11-06, 2022-11-07, 2022-11-08,
2022-11-09, 2022-11-10, 2022-11-11, 2022-11-12, 2022-11-13, 2022-11-14,
2022-11-15, 2022-11-16, 2022-11-17, 2022-11-18, 2022-11-19, 2022-11-20,
2022-11-21, 2022-11-22, 2022-11-23, 2022-11-24, 2022-11-25, 2022-11-26,
2022-11-27, 2022-11-28, 2022-11-29, 2022-11-30, 2022-12-01, 2022-12-02,
2022-12-03, 2022-12-04, 2022-12-05, 2022-12-06, 2022-12-07, 2022-12-08,
2022-12-09, 2022-12-10, 2022-12-11, 2022-12-12, 2022-12-13, 2022-12-14,
2022-12-15, 2022-12-16, 2022-12-17, 2022-12-18, 2022-12-19, 2022-12-20,
2022-12-21, 2022-12-22, 2022-12-23, 2022-12-24, 2022-12-25, 2022-12-26,
2022-12-27, 2022-12-28, 2022-12-29, 2022-12-30, 2022-12-31, 2023-01-01,
2023-01-02, 2023-01-03, 2023-01-04, 2023-01-05, 2023-01-06, 2023-01-07,
2023-01-08, 2023-01-09, 2023-01-10, 2023-01-11, 2023-01-12, 2023-01-13,
2023-01-14, 2023-01-15, 2023-01-16, 2023-01-17, 2023-01-18, 2023-01-19,
2023-01-20, 2023-01-21, 2023-01-22, 2023-01-23, 2023-01-24, 2023-01-25,
2023-01-26, 2023-01-27, 2023-01-28, 2023-01-29, 2023-01-30, 2023-01-31,
2023-02-01, 2023-02-02, 2023-02-03, 2023-02-04, 2023-02-05, 2023-02-06,
2023-02-07, 2023-02-08, 2023-02-09, 2023-02-10, 2023-02-11, 2023-02-12,
2023-02-13, 2023-02-14, 2023-02-15, 2023-02-16, 2023-02-17, 2023-02-18,
2023-02-19, 2023-02-20, 2023-02-21, 2023-02-22, 2023-02-23, 2023-02-24,
2023-02-25, 2023-02-26, 2023-02-27, 2023-02-28, 2023-03-01, 2023-03-02,
2023-03-03, 2023-03-04, 2023-03-05, 2023-03-06, 2023-03-07, 2023-03-08,
2023-03-09, 2023-03-10, 2023-03-11, 2023-03-12, 2023-03-13, 2023-03-14,
2023-03-15, 2023-03-16, 2023-03-17, 2023-03-18, 2023-03-19, 2023-03-20,
2023-03-21, 2023-03-22, 2023-03-23, 2023-03-24, 2023-03-25, 2023-03-26,
2023-03-27, 2023-03-28, 2023-03-29, 2023-03-30, 2023-03-31, 2023-04-01,
2023-04-02, 2023-04-03, 2023-04-04, 2023-04-05, 2023-04-06, 2023-04-07,
2023-04-08, 2023-04-09, 2023-04-10, 2023-04-11, 2023-04-12, 2023-04-13,
2023-04-14, 2023-04-15, 2023-04-16, 2023-04-17, 2023-04-18, 2023-04-19,
2023-04-20, 2023-04-21, 2023-04-22, 2023-04-23, 2023-04-24, 2023-04-25,
2023-04-26, 2023-04-27, 2023-04-28, 2023-04-29, 2023-04-30, 2023-05-01,
2023-05-02, 2023-05-03, 2023-05-04, 2023-05-05, 2023-05-06, 2023-05-07,
2023-05-08, 2023-05-09, 2023-05-10, 2023-05-11, 2023-05-12, 2023-05-13,
2023-05-14, 2023-05-15, 2023-05-16, 2023-05-17, 2023-05-18, 2023-05-19,
2023-05-20, 2023-05-21, 2023-05-22, 2023-05-23, 2023-05-24, 2023-05-25,
2023-05-26, 2023-05-27, 2023-05-28, 2023-05-29, 2023-05-30, 2023-05-31}
```

To adjust to the population size of the province:

```
In[∘]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "QC"][[1, 1]]]]
Out[∘]= {QC, 8 695 659, 1 250 508}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[∘]:= per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
        per100000A = 100 000 / popsize[[3]]
Out[∘]= 100 000 / 7 445 151
Out[∘]= 25 000 / 312 627
```

```
In[∘]:= Clear[datetabQC];
        datetabQC[i_] = {0, 0};
        For[i = 1, i ≤ Length[counts], i++,
         date = Position[dates, counts[[i, 1]]][[1, 1]];
         If[((counts[[i, 2]] == "70-79 ans") ||
             (counts[[i, 2]] == "80-89 ans") || (counts[[i, 2]] == "90 ans et plus")),
          datetabQC[date] = datetabQC[date] + {0, per100000A * counts[[i, 3]]},
          datetabQC[date] = datetabQC[date] + {per100000Y * counts[[i, 3]], 0}
         ]
        ]
```

Processing QC Case Data [ENTER this or the previous folder - pre-processed data]

```
In[∘]:= startafter = "2021-11-30";
```

The two periods we’ll consider:

```
In[∘]:= mystartdate1 = "2021-12-01";
        myenddate1 = "2022-08-31";
        mystartdate2 = "2022-09-01";
        myenddate2 = "2023-05-31";
```

```
In[∘]:= curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

```
In[•]:= xaxis = {{curdate["2021-10-01"], "Oct"},
    {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
    {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
    {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
    {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
    {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
    {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
    {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
    {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
    {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
    {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

To adjust to the population size of the province:

```
In[•]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "QC"][[1, 1]]]]
Out[•]= {QC, 8695659, 1250508}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[•]:= per100000Y = 100000 / (popsize[[2]] - popsize[[3]])
        per100000A = 100000 / popsize[[3]]
```

Out[•]= $\frac{100000}{7445151}$

Out[•]= $\frac{25000}{312627}$

```
In[•]:= tabdates = {"2021-12-01", "2021-12-02", "2021-12-03",
    "2021-12-04", "2021-12-05", "2021-12-06", "2021-12-07",
    "2021-12-08", "2021-12-09", "2021-12-10", "2021-12-11", "2021-12-12",
    "2021-12-13", "2021-12-14", "2021-12-15", "2021-12-16", "2021-12-17",
    "2021-12-18", "2021-12-19", "2021-12-20", "2021-12-21", "2021-12-22",
    "2021-12-23", "2021-12-24", "2021-12-25", "2021-12-26", "2021-12-27",
    "2021-12-28", "2021-12-29", "2021-12-30", "2021-12-31", "2022-01-01",
    "2022-01-02", "2022-01-03", "2022-01-04", "2022-01-05", "2022-01-06",
    "2022-01-07", "2022-01-08", "2022-01-09", "2022-01-10", "2022-01-11",
    "2022-01-12", "2022-01-13", "2022-01-14", "2022-01-15", "2022-01-16",
    "2022-01-17", "2022-01-18", "2022-01-19", "2022-01-20", "2022-01-21",
    "2022-01-22", "2022-01-23", "2022-01-24", "2022-01-25", "2022-01-26",
    "2022-01-27", "2022-01-28", "2022-01-29", "2022-01-30", "2022-01-31",
    "2022-02-01", "2022-02-02", "2022-02-03", "2022-02-04", "2022-02-05",
    "2022-02-06", "2022-02-07", "2022-02-08", "2022-02-09", "2022-02-10",
    "2022-02-11", "2022-02-12", "2022-02-13", "2022-02-14", "2022-02-15",
    "2022-02-16", "2022-02-17", "2022-02-18", "2022-02-19", "2022-02-20",
    "2022-02-21", "2022-02-22", "2022-02-23", "2022-02-24", "2022-02-25",
```

```
"2022-02-26", "2022-02-27", "2022-02-28", "2022-03-01", "2022-03-02",
"2022-03-03", "2022-03-04", "2022-03-05", "2022-03-06", "2022-03-07",
"2022-03-08", "2022-03-09", "2022-03-10", "2022-03-11", "2022-03-12",
"2022-03-13", "2022-03-14", "2022-03-15", "2022-03-16", "2022-03-17",
"2022-03-18", "2022-03-19", "2022-03-20", "2022-03-21", "2022-03-22",
"2022-03-23", "2022-03-24", "2022-03-25", "2022-03-26", "2022-03-27",
"2022-03-28", "2022-03-29", "2022-03-30", "2022-03-31", "2022-04-01",
"2022-04-02", "2022-04-03", "2022-04-04", "2022-04-05", "2022-04-06",
"2022-04-07", "2022-04-08", "2022-04-09", "2022-04-10", "2022-04-11",
"2022-04-12", "2022-04-13", "2022-04-14", "2022-04-15", "2022-04-16",
"2022-04-17", "2022-04-18", "2022-04-19", "2022-04-20", "2022-04-21",
"2022-04-22", "2022-04-23", "2022-04-24", "2022-04-25", "2022-04-26",
"2022-04-27", "2022-04-28", "2022-04-29", "2022-04-30", "2022-05-01",
"2022-05-02", "2022-05-03", "2022-05-04", "2022-05-05", "2022-05-06",
"2022-05-07", "2022-05-08", "2022-05-09", "2022-05-10", "2022-05-11",
"2022-05-12", "2022-05-13", "2022-05-14", "2022-05-15", "2022-05-16",
"2022-05-17", "2022-05-18", "2022-05-19", "2022-05-20", "2022-05-21",
"2022-05-22", "2022-05-23", "2022-05-24", "2022-05-25", "2022-05-26",
"2022-05-27", "2022-05-28", "2022-05-29", "2022-05-30", "2022-05-31",
"2022-06-01", "2022-06-02", "2022-06-03", "2022-06-04", "2022-06-05",
"2022-06-06", "2022-06-07", "2022-06-08", "2022-06-09", "2022-06-10",
"2022-06-11", "2022-06-12", "2022-06-13", "2022-06-14", "2022-06-15",
"2022-06-16", "2022-06-17", "2022-06-18", "2022-06-19", "2022-06-20",
"2022-06-21", "2022-06-22", "2022-06-23", "2022-06-24", "2022-06-25",
"2022-06-26", "2022-06-27", "2022-06-28", "2022-06-29", "2022-06-30",
"2022-07-01", "2022-07-02", "2022-07-03", "2022-07-04", "2022-07-05",
"2022-07-06", "2022-07-07", "2022-07-08", "2022-07-09", "2022-07-10",
"2022-07-11", "2022-07-12", "2022-07-13", "2022-07-14", "2022-07-15",
"2022-07-16", "2022-07-17", "2022-07-18", "2022-07-19", "2022-07-20",
"2022-07-21", "2022-07-22", "2022-07-23", "2022-07-24", "2022-07-25",
"2022-07-26", "2022-07-27", "2022-07-28", "2022-07-29", "2022-07-30",
"2022-07-31", "2022-08-01", "2022-08-02", "2022-08-03", "2022-08-04",
"2022-08-05", "2022-08-06", "2022-08-07", "2022-08-08", "2022-08-09",
"2022-08-10", "2022-08-11", "2022-08-12", "2022-08-13", "2022-08-14",
"2022-08-15", "2022-08-16", "2022-08-17", "2022-08-18", "2022-08-19",
"2022-08-20", "2022-08-21", "2022-08-22", "2022-08-23", "2022-08-24",
"2022-08-25", "2022-08-26", "2022-08-27", "2022-08-28", "2022-08-29",
"2022-08-30", "2022-08-31", "2022-09-01", "2022-09-02", "2022-09-03",
"2022-09-04", "2022-09-05", "2022-09-06", "2022-09-07", "2022-09-08",
"2022-09-09", "2022-09-10", "2022-09-11", "2022-09-12", "2022-09-13",
"2022-09-14", "2022-09-15", "2022-09-16", "2022-09-17", "2022-09-18",
"2022-09-19", "2022-09-20", "2022-09-21", "2022-09-22", "2022-09-23",
"2022-09-24", "2022-09-25", "2022-09-26", "2022-09-27", "2022-09-28",
"2022-09-29", "2022-09-30", "2022-10-01", "2022-10-02", "2022-10-03",
```

```
"2022-10-04", "2022-10-05", "2022-10-06", "2022-10-07", "2022-10-08",
"2022-10-09", "2022-10-10", "2022-10-11", "2022-10-12", "2022-10-13",
"2022-10-14", "2022-10-15", "2022-10-16", "2022-10-17", "2022-10-18",
"2022-10-19", "2022-10-20", "2022-10-21", "2022-10-22", "2022-10-23",
"2022-10-24", "2022-10-25", "2022-10-26", "2022-10-27", "2022-10-28",
"2022-10-29", "2022-10-30", "2022-10-31", "2022-11-01", "2022-11-02",
"2022-11-03", "2022-11-04", "2022-11-05", "2022-11-06", "2022-11-07",
"2022-11-08", "2022-11-09", "2022-11-10", "2022-11-11", "2022-11-12",
"2022-11-13", "2022-11-14", "2022-11-15", "2022-11-16", "2022-11-17",
"2022-11-18", "2022-11-19", "2022-11-20", "2022-11-21", "2022-11-22",
"2022-11-23", "2022-11-24", "2022-11-25", "2022-11-26", "2022-11-27",
"2022-11-28", "2022-11-29", "2022-11-30", "2022-12-01", "2022-12-02",
"2022-12-03", "2022-12-04", "2022-12-05", "2022-12-06", "2022-12-07",
"2022-12-08", "2022-12-09", "2022-12-10", "2022-12-11", "2022-12-12",
"2022-12-13", "2022-12-14", "2022-12-15", "2022-12-16", "2022-12-17",
"2022-12-18", "2022-12-19", "2022-12-20", "2022-12-21", "2022-12-22",
"2022-12-23", "2022-12-24", "2022-12-25", "2022-12-26", "2022-12-27",
"2022-12-28", "2022-12-29", "2022-12-30", "2022-12-31", "2023-01-01",
"2023-01-02", "2023-01-03", "2023-01-04", "2023-01-05", "2023-01-06",
"2023-01-07", "2023-01-08", "2023-01-09", "2023-01-10", "2023-01-11",
"2023-01-12", "2023-01-13", "2023-01-14", "2023-01-15", "2023-01-16",
"2023-01-17", "2023-01-18", "2023-01-19", "2023-01-20", "2023-01-21",
"2023-01-22", "2023-01-23", "2023-01-24", "2023-01-25", "2023-01-26",
"2023-01-27", "2023-01-28", "2023-01-29", "2023-01-30", "2023-01-31",
"2023-02-01", "2023-02-02", "2023-02-03", "2023-02-04", "2023-02-05",
"2023-02-06", "2023-02-07", "2023-02-08", "2023-02-09", "2023-02-10",
"2023-02-11", "2023-02-12", "2023-02-13", "2023-02-14", "2023-02-15",
"2023-02-16", "2023-02-17", "2023-02-18", "2023-02-19", "2023-02-20",
"2023-02-21", "2023-02-22", "2023-02-23", "2023-02-24", "2023-02-25",
"2023-02-26", "2023-02-27", "2023-02-28", "2023-03-01", "2023-03-02",
"2023-03-03", "2023-03-04", "2023-03-05", "2023-03-06", "2023-03-07",
"2023-03-08", "2023-03-09", "2023-03-10", "2023-03-11", "2023-03-12",
"2023-03-13", "2023-03-14", "2023-03-15", "2023-03-16", "2023-03-17",
"2023-03-18", "2023-03-19", "2023-03-20", "2023-03-21", "2023-03-22",
"2023-03-23", "2023-03-24", "2023-03-25", "2023-03-26", "2023-03-27",
"2023-03-28", "2023-03-29", "2023-03-30", "2023-03-31", "2023-04-01",
"2023-04-02", "2023-04-03", "2023-04-04", "2023-04-05", "2023-04-06",
"2023-04-07", "2023-04-08", "2023-04-09", "2023-04-10", "2023-04-11",
"2023-04-12", "2023-04-13", "2023-04-14", "2023-04-15", "2023-04-16",
"2023-04-17", "2023-04-18", "2023-04-19", "2023-04-20", "2023-04-21",
"2023-04-22", "2023-04-23", "2023-04-24", "2023-04-25", "2023-04-26",
"2023-04-27", "2023-04-28", "2023-04-29", "2023-04-30", "2023-05-01",
"2023-05-02", "2023-05-03", "2023-05-04", "2023-05-05", "2023-05-06",
"2023-05-07", "2023-05-08", "2023-05-09", "2023-05-10", "2023-05-11",
```

```
"2023-05-12", "2023-05-13", "2023-05-14", "2023-05-15", "2023-05-16",
"2023-05-17", "2023-05-18", "2023-05-19", "2023-05-20", "2023-05-21",
"2023-05-22", "2023-05-23", "2023-05-24", "2023-05-25", "2023-05-26",
"2023-05-27", "2023-05-28", "2023-05-29", "2023-05-30", "2023-05-31"};
```

In[*]:=
```
tab = {{16.48052537819582`, 5.037952576073084`},
  {18.857911679695953`, 4.7180825712430465`}, {17.971428652017938`,
   4.318245065205501`}, {15.99698918128054`, 5.357822580903121`},
  {14.895601177195735`, 3.4386025519228984`}, {16.44023069511955`,
   5.43779008211063`}, {19.95929968378076`, 5.597725084525649`},
  {25.358787216001396`, 7.836815118335909`}, {25.37221877702682`,
   7.277042609883344`}, {25.94977590112007`, 7.6768801159208895`},
  {22.000896959645278`, 6.157497592978213`}, {21.22186642017066`,
   4.957985074865575`}, {27.763036639552375`, 7.756847617128399`},
  {34.38479622508664`, 8.956360135241038`}, {43.397373673146454`,
   10.235840154561187`}, {52.22190926685033`, 11.835190178711372`},
  {55.41862079090135`, 15.833565239086836`}, {60.32114056518128`,
   16.953110255991966`}, {56.69461908831668`, 14.634052720974196`},
  {85.1963915842674`, 19.112232788594714`}, {107.55994069159914`,
   21.83112782965003`}, {148.56649650222005`, 27.029015408138132`},
  {152.6765541759999`, 33.82625301077642`}, {151.2931033903812`,
   30.227715456438503`}, {138.07644734136352`, 24.1501853646678`},
  {135.6587663567871`, 32.30687048783374`}, {173.22684254489937`,
   43.90215816292259`}, {197.96777795373123`, 58.37627588148177`},
  {240.12944801253863`, 71.6508810819283`}, {238.51766068948768`,
   73.88997111573856`}, {214.94527108986776`, 78.92792369181164`},
  {184.81827970984068`, 82.8463312509796`}, {176.4369856299758`,
   79.3277611978492`}, {176.23551221459445`, 72.69045859762592`},
  {220.0761274015799`, 101.7986290371593`}, {220.15671676773243`,
   121.95043934145163`}, {201.2988050880365`, 113.31394921104064`},
  {165.53055807733114`, 117.63219427624614`}, {108.41956059722631`,
   97.00057896470874`}, {96.27742943024258`, 84.36571377392228`},
  {85.77394870836065`, 82.52646124614957`}, {80.24014556588577`,
   117.3123242714161`}, {72.91994480702944`, 118.59180429073625`},
  {69.23969708606313`, 97.64031897436881`}, {59.8913306123677`,
   84.28574627271477`}, {55.86186230474036`, 83.48607126063968`},
  {47.399978858722946`, 58.136373377859236`}, {51.442878727375714`,
   61.17513842374459`}, {57.79600709240148`, 72.21065359038087`},
  {54.733611178604704`, 68.53214853483544`}, {48.90431369357049`,
   70.13149855898563`}, {42.40343815726504`, 69.41179104811805`},
  {34.680290567645976`, 53.25835580420117`}, {31.335831872315282`,
   33.82625301077642`}, {40.52301961370562`, 35.665505538549134`},
  {47.211937004367`, 52.77855079695612`}, {41.30205015318024`,
   47.660630719675524`}, {39.58281034192591`, 52.37871329091857`},
```

```
{37.24571872350205`, 38.46436808081196`}, {31.953683679484808`,
 24.629990371912854`}, {29.777770793366045`, 23.91028286104527`},
{37.63523399323936`, 28.22852792625077`}, {44.45846699415499`,
 39.90378310254712`}, {41.85274415522264`, 33.10654549990884`},
{34.438522469188335`, 27.748722919005715`}, {34.0355756384256`,
 27.588787916590697`}, {26.164680877526862`, 23.350510352592707`},
{24.700640725755594`, 21.75116032844252`}, {35.190689886612105`,
 25.10979537915791`}, {35.62049983942569`, 32.94661049749382`},
{31.040337529755945`, 28.62836543228832`}, {28.26000439749308`,
 25.509632885195458`}, {24.714072286781022`, 23.11060784897018`},
{18.105744262272182`, 17.43291526323702`}, {18.69673294739086`,
 14.554085219766687`}, {23.075421841679237`, 17.272980260822003`},
{23.223169012958905`, 20.231777805499846`}, {22.18893881400122`,
 19.35213529221724`}, {19.999594366857032`, 17.67281776685955`},
{16.53425162229752`, 19.67200529704728`}, {12.86743546235664`,
 16.953110255991966`}, {13.565876635678713`, 13.594475205276575`},
{19.072816656102745`, 15.913532740294345`}, {18.119175823297606`,
 22.790737844140143`}, {16.60140942742464`, 17.43291526323702`},
{14.61353839566182`, 18.152622774104604`}, {11.537710920839618`,
 14.793987723389215`}, {11.081037845975185`, 10.075905152146168`},
{10.275144184449717`, 11.595287675088844`}, {14.532949029509274`,
 16.47330524874691`}, {13.13606668286513`, 17.67281776685955`},
{12.84057234030579`, 14.873955224596724`}, {13.673329123882107`,
 15.993500241501854`}, {11.295942822381978`, 16.3933377475394`},
{9.10659837523779`, 10.315807655768696`}, {9.120029936263213`,
 8.396587626788474`}, {14.801580250017762`, 15.19382522942676`},
{15.446295179238138`, 14.07428021252163`}, {13.418129464399042`,
 18.55246028014215`}, {12.263015216212539`, 15.593662735464306`},
{11.175058773153157`, 13.354572701654048`}, {8.260410030636047`,
 10.395775156976205`}, {7.951484127051285`, 10.155872653353677`},
{14.304612492077059`, 18.952297786179695`}, {15.99698918128054`,
 18.792362783764677`}, {16.762588159729734`, 20.471680309122373`},
{14.922464299246583`, 19.032265287387204`}, {12.948024828509187`,
 19.592037795839772`}, {10.772111942390422`, 11.515320173881335`},
{12.289878338263387`, 13.674442706484085`}, {20.039889049933304`,
 17.193012759614493`}, {26.31242804880653`, 25.34969788278044`},
{27.803331322628647`, 24.470055369497835`}, {25.56026063138276`,
 23.670380357422744`}, {24.714072286781022`, 23.750347858630253`},
{18.468396409958643`, 20.87151781515992`}, {20.429404319670613`,
 19.91190780066981`}, {33.807239100993385`, 30.307682957646012`},
{36.587572233256246`, 33.42641550473888`}, {41.46322888548533`,
 37.1049205602843`}, {34.948921788154465`, 39.6638805989246`},
{33.59233412458659`, 40.22365310737716`}, {26.12438619445059`,
 35.74547303975664`}, {25.76173404676413`, 29.747910449193448`},
```

{38.5351485819428`, 45.581475688280285`}, {42.56461688957014`,
47.98050072450556`}, {39.39476848756996`, 50.61942826435337`},
{38.3739698496377`, 59.255918394764365`}, {30.986611285654245`,
53.57822580903121`}, {26.285564926755683`, 53.17838830299366`},
{25.211040044721727`, 46.86095570760043`}, {32.07456772871363`,
43.34238565447002`}, {36.815908770688466`, 66.85283100947775`},
{35.3518686189172`, 66.93279851068526`}, {29.495708011832132`,
59.97562590563195`}, {23.048558719628385`, 49.97968825469329`},
{18.18633362842473`, 36.30524554820921`}, {16.587977866399218`,
31.74709797938118`}, {22.229233497077495`, 35.505570536134115`},
{30.167286063103354`, 43.262418153262516`}, {30.691116943094908`,
53.89809581386125`}, {25.466239704204792`, 41.58310062790482`},
{21.73226573913679`, 41.18326312186727`}, {14.492654346433001`,
28.14856042504326`}, {15.177663958729648`, 26.14937289485553`},
{20.630877735051982`, 35.98537554337917`}, {20.483130563772313`,
34.94579802768155`}, {18.226628311501003`, 36.30524554820921`},
{17.64907118740775`, 36.465180550624225`}, {14.10313907669569`,
31.507195475758653`}, {10.7049541372633`, 19.91190780066981`},
{9.32150335164458`, 18.152622774104604`}, {16.19846259666191`,
27.668755417798206`}, {16.064146986407664`, 31.9070329817962`},
{14.13000219874654`, 20.231777805499846`}, {12.061541800831172`,
28.14856042504326`}, {10.664659454187028`, 20.471680309122373`},
{7.118727343474967`, 16.47330524874691`}, {7.4948110521868525`,
13.034702696824011`}, {10.288575745475143`, 16.07346774270936`},
{11.282511261356552`, 22.630802841725124`}, {9.912492036763258`,
16.23340274512438`}, {8.44845188499199`, 18.312557776519622`},
{7.8440316388478895`, 15.833565239086836`}, {6.097928705542708`,
12.0750926823339`}, {5.748708118881672`, 8.876392634033529`},
{7.30676919783091`, 15.513695234256797`}, {7.655989784491947`,
11.195450169051298`}, {6.957548611169874`, 9.995937650938659`},
{5.896455290161342`, 9.516132643693602`}, {5.533803142474881`,
9.356197641278584`}, {3.881721136347671`, 8.236652624373455`},
{4.257804845059556`, 7.117107607468325`}, {6.447149292203744`,
7.676880115920889 5`}, {7.5351057352631265`, 9.67606764610862`},
{7.4948110521868525`, 11.195450169051298`}, {6.4874439752800175`,
8.396587626788474`}, {5.748708118881672`, 8.396587626788474`},
{4.593593870695168`, 7.357010111090853`}, {4.620456992746017`,
6.317432595393232`}, {7.629126662441098`, 9.83600264852364`},
{8.193252225508925`, 8.636490130411001`}, {9.13346149728863 7`,
9.196262638863566`}, {6.648622707585112`, 8.79642513282602`},
{5.399487532220636`, 8.076717621958435`}, {5.036835384534175`,
5.997562590563195`}, {5.1845825558138445`, 5.597725084525649`},
{8.891693398830997`, 10.87558016422126`}, {8.1798206644835`,
9.516132643693602`}, {9.160324619339487`, 10.555710159391223`},

{8.555904373195386`, 9.995937650938659`}, {6.662054268610536`, 10.71564516180624`}, {5.251740360940967`, 10.155872653353677`}, {5.735276557856247`, 8.476555127995983`}, {9.83190267061071`, 11.195450169051298`}, {9.402092717797128`, 11.115482667843787`}, {10.85270130854297`, 13.514507704069066`}, {9.59013457215307`, 14.793987723389215`}, {8.354430957814019`, 11.675255176296353`}, {7.105295782449543`, 10.315807655768696`}, {6.648622707585112`, 9.116295137656056`}, {12.14213166983718`, 14.793987723389215`}, {13.552445074653289`, 13.034702696824011`}, {15.204527080780498`, 17.832752769274567`}, {11.833205263398956`, 18.152622774104604`}, {9.912492036763258`, 14.793987723389215`}, {8.864830276780149`, 13.514507704069066`}, {9.549839889076797`, 12.314995185956427`}, {18.40123860483152`, 21.111420318782447`}, {19.27429007148411`, 26.869080405723114`}, {18.64300670328916`, 26.949047906930623`}, {17.64907118740775`, 27.18895041055315`}, {11.954089312627776`, 24.550022870705345`}, {11.430258432636222`, 23.510445355007725`}, {11.725752775195561`, 21.431290323612483`}, {22.565022522713104`, 31.267292972136122`}, {19.623510658145147`, 35.4256030349266`}, {20.95323519966217`, 36.865018056661775`}, {17.957997090992514`, 36.625115553039244`}, {15.298548007958468`, 31.027390468513595`}, {11.658594970068437`, 29.268105441948393`}, {12.128699605958294`, 27.26891791176066`}, {20.53685680787401`, 38.784238085641995`}, {22.04119164272155`, 44.94173567862021`}, {23.948473308331824`, 44.14206066654511`}, {19.39517412071293`, 37.584725567529354`}, {19.945868122755336`, 41.4231656254898`}, {15.070211470526253`, 32.30687048783374`}, {14.667264639763518`, 24.789925374327872`}, {22.269528180153767`, 38.94417308805701`}, {25.869186534967525`, 38.06453057477441`}, {21.933739154518154`, 35.74547303975664`}, {20.066752171984156`, 36.465180550624225`}, {16.077578547433088`, 32.62674049266378`}, {12.437625509543057`, 28.54839793108081`}, {12.92116170645833 9`, 25.269730381572927`}, {18.37437548278067`, 34.545960521644005`}, {17.91770240791624`, 37.424790565114336`}, {15.876105132051721`, 31.347260473343635`}, {14.667264639763518`, 29.42804044436341`}, {12.854003901331215`, 30.62755296247605`}, {9.818471109585285`, 19.991875301877318`}, {9.20061930241576`, 22.790737844140143`}, {13.847939417212626`, 28.30849542745828`}, {13.565876635678713`, 34.94579802768155`}, {13.37783478132277`, 27.588787916590697`}, {9.657292377280193`, 25.829502890025495`}, {9.818471109585285`, 29.268105441948393`}, {7.333632319881759`, 19.192200289802223`}, {7.42765324705973`, 18.712395282557168`}, {12.854003901331215`, 25.10979537915791`}, {11.672026531093863`, 21.351322822404974`}, {11.60486872596674`, 24.390087868290326`}, {11.134764090076883`, 29.108170439533374`}, {8.864830276780149`, 20.7115828127449`}, {7.42765324705973`, 18.072655272897094`},

{6.756075195788507`, 16.953110255991966`}, {11.497416237763344`,
29.42804044436341`}, {11.860068385449805`, 30.38765045885352`},
{11.228785017254856`, 22.95067284655516`}, {9.402092717797128`,
23.830315359837762`}, {8.246978469610623`, 19.991875301877318`},
{6.218812754771529`, 17.912720270482076`}, {6.111360266568132`,
15.27379273063427`}, {9.509545206000523`, 19.192200289802223`},
{10.140828574195472`, 21.5912253260275`}, {8.757377788576752`,
20.231777805499846`}, {7.373927002958033`, 17.67281776685955`},
{7.750010711669918`, 14.714020222181706`}, {5.426350654271485`,
13.354572701654048`}, {4.78163572505111`, 10.635677660598732`},
{8.005210371152982`, 16.63324025116193`}, {9.670723938305617`,
18.952297786179695`}, {8.340999396788595`, 17.832752769274567`},
{7.548537296288551`, 19.192200289802223`}, {6.1650865106698305`,
16.31337024633189`}, {4.553299187618895`, 11.115482667843787`},
{4.257804845059556`, 10.95554766542877`}, {6.272538998873227`,
11.835190178711372`}, {9.805039548559861`, 14.953922725804233`},
{8.636493739347932`, 17.912720270482076`}, {9.07973525318694`,
15.513695234256797`}, {6.406854609127471`, 13.994312711314121`},
{5.600960947602003`, 12.954735195616502`}, {5.251740360940967`,
13.994312711314121`}, {9.173756180364911`, 18.872330284972186`},
{8.609630617297084`, 16.3933377475394`}, {8.126094420381802`,
13.834377708899103`}, {7.521674174237702`, 15.433727733049288`},
{5.869592168110493`, 15.513695234256797`}, {4.445846699415498`,
10.635677660598732`}, {5.68155031375455`, 14.23421521493665`},
{9.281208668568308`, 18.872330284972186`}, {8.663356861398782`,
21.431290323612483`}, {8.488746568068263`, 17.43291526323702`},
{8.327567835763169`, 19.75197279825479`}, {7.42765324705973`,
17.352947762029512`}, {5.762139679907096`, 16.55327274995442`},
{5.80243436298337`, 13.674442706484085`}, {9.899060475737832`,
18.63242778134966`}, {10.436322916754811`, 24.390087868290326`},
{10.140828574195472`, 22.071030333272557`}, {8.273841591661473`,
22.150997834480066`}, {8.582767495246236`, 19.432102793424754`},
{5.103993189661297`, 17.832752769274567`}, {6.43371773117832`,
19.512070294632263`}, {9.07973525318694`, 22.310932836895084`},
{10.288575745475143`, 27.18895041055315`}, {9.308071790619156`,
22.710770342932634`}, {8.95885120395812`, 29.108170439533374`},
{8.488746568068263`, 23.91028286104527`}, {5.359192849144362`,
18.312557776519622`}, {5.721844996830823`, 22.390900338102597`},
{7.132158904500392`, 19.91190780066981`}, {10.637796332136178`,
24.709957873120363`}, {11.24221657828028`, 31.98700048300371`},
{9.805039548559861`, 26.22934039606304`}, {9.146893058314062`,
26.789112904515605`}, {5.735276557856247`, 22.550835340517615`},
{5.641255630678276`, 18.232590275312113`}, {9.966218280864956`,
24.86989287553538`}, {9.85876579266156`, 27.748722919005715`},

```
{9.4423874008734`, 25.909470391233004`}, {8.354430957814019`,
 22.630802841725124`}, {6.608328024508838`, 21.5912253260275`},
{5.023403823508751`, 17.352947762029512`}, {5.278603482991816`,
 16.23340274512438`}, {9.643860816254767`, 27.828690420213224`},
{9.173756180364911`, 26.22934039606304`}, {8.636493739347932`,
 27.668755417798206`}, {7.964915688076709`, 27.26891791176066`},
{7.011274855271572`, 24.94986037674289`}, {4.405552016339225`,
 17.67281776685955`}, {5.023403823508751`, 14.714020222181706`},
{8.461883446017415`, 21.191387819989956`}, {9.549839889076797`,
 25.749535388817986`}, {7.884326321924163`, 22.310932836895084`},
{8.193252225508925`, 25.829502890025495`}, {7.279906075780062`,
 21.75116032844252`}, {4.915951335305355`, 19.112232788594714`},
{5.265171921966392`, 19.512070294632263`}, {8.340999396788595`,
 24.390087868290326`}, {8.287273152686897`, 23.430477853800216`},
{8.139525981407226`, 22.310932836895084`}, {6.232244315796953`,
 20.951485316367428`}, {5.896455290161342`, 18.63242778134966`},
{4.230941723008708`, 16.31337024633189`}, {4.083194551729038`,
 17.51288276444453`}, {7.5754004183394`, 19.991875301877318`},
{9.173756180364911`, 27.828690420213224`}, {7.226179831678364`,
 24.07021786346029`}, {6.635191146559687`, 23.99025036225278`},
{5.990476217339313`, 21.031452817574937`}, {5.305466605042665`,
 17.193012759614493`}, {5.117424750686722`, 18.952297786179695`},
{9.052872131136091`, 23.830315359837762`}, {9.791607987534437`,
 27.588787916590697`}, {9.791607987534437`, 23.99025036225278`},
{7.884326321924163`, 28.06859292383575`}, {7.30676919783091`,
 24.709957873120363`}, {5.762139679907096`, 21.75116032844252`},
{6.124791827593557`, 23.510445355007725`}, {10.745248820339574`,
 28.30849542745828`}, {11.040743162898913`, 37.1049205602843`},
{11.927226190576928`, 34.06615551439895`}, {9.187187741390336`,
 24.07021786346029`}, {8.784240910627602`, 27.18895041055315`},
{6.702348951686809`, 24.310120367082817`}, {6.339696804000349`,
 19.75197279825479`}, {10.798975064441272`, 31.267292972136122`},
{12.007815556729474`, 36.30524554820921`}, {11.067606284949761`,
 32.226902986626236`}, {9.885628914712408`, 29.188137940740884`},
{8.05893661525468`, 29.188137940740884`}, {5.990476217339313`,
 29.987812952815975`}, {6.3128336819495`, 23.510445355007725`},
{11.175058773153157`, 26.709145403308096`}, {11.645163409043013`,
 35.665505538549134`}, {11.940657751602352`, 37.824628071151885`},
{10.436322916754811`, 35.3456355337191`}, {8.56933593422081`,
 33.746285509568914`}, {7.1858851486020905`, 26.14937289485553`},
{6.890390806042752`, 29.348072943155902`}, {11.255648139305704`,
 35.82544054096415`}, {12.84057234030579`, 40.143685606169655`},
{10.839269747517545`, 32.62674049266378`}, {8.367862518839443`,
 29.42804044436341`}, {5.668118752729125`, 27.34888541296817`},
```

```
{3.814563331220549`, 18.232590275312113`}, {2.8877856204662606`,
 20.391712807914864`}, {5.694981874779974`, 27.508820415383187`},
{8.139525981407226`, 29.108170439533374`}, {9.95278671983953`,
 31.98700048300371`}, {8.69021998344963`, 44.54189817258266`},
{6.635191146559687`, 25.909470391233004`}, {4.53986762659347`,
 21.75116032844252`}, {2.874354059440836`, 19.032265287387204`},
{6.568033341432565`, 24.310120367082817`}, {8.139525981407226`,
 25.829502890025495`}, {9.214050863441186`, 32.7866754950788`},
{6.930685489119025`, 29.42804044436341`}, {5.990476217339313`,
 25.909470391233004`}, {4.284667967110405`, 18.47249277893464`},
{4.042899868652765`, 22.630802841725124`}, {6.299402120924075`,
 25.749535388817986`}, {5.654687191703701`, 22.870705345347652`},
{5.265171921966392`, 27.988625422628242`}, {4.647320114796866`,
 22.870705345347652`}, {3.4519111835340883`, 18.47249277893464`},
{3.304164012254419`, 16.15343524391687`}, {3.7742686481442753`,
 16.15343524391687`}, {5.29203504401724`, 20.87151781515992`},
{5.29203504401724`, 19.432102793424754`}, {4.271236406084981`,
 18.792362783764677`}, {3.935447380449369`, 18.952297786179695`},
{3.0355327917459296`, 18.152622774104604`}, {2.6863122050848935`,
 9.436165142486093`}, {2.6191543999577713`, 11.355385171466317`},
{4.741341041974837`, 14.554085219766687`}, {4.78163572505111`,
 13.994312711314121`}, {5.023403823508751`, 10.95554766542877`},
{3.8414264532713975`, 10.635677660598732`}, {3.6936792819917286`,
 11.275417670258808`}, {2.699743766110318`, 9.196262638863566`},
{3.0086696696950805`, 9.116295137656056`}, {5.090561628635873`,
 12.394962687163936`}, {5.399487532220636`, 14.394150217351669`},
{5.063698506585024`, 13.19463769923903`}, {4.808498847101959`,
 13.914345210106612`}, {3.532500549686635`, 11.595287675088844`},
{2.4042494235509797`, 9.356197641278584`}, {3.196711524051023`,
 11.99512518112639`}, {5.359192849144362`, 15.113857728219251`},
{5.439782215296909`, 16.793175253576948`}, {5.305466605042665`,
 15.593662735464306`}, {4.674183236847715`, 14.873955224596724`},
{4.1772154789070095`, 15.593662735464306`}, {3.15641684097475`,
 13.354572701654048`}, {2.780333132262865`, 13.754410207691594`},
{5.426350654271485`, 14.793987723389215`}, {5.224877238890118`,
 15.673630236671817`}, {4.848793530178233`, 15.433727733049288`},
{4.230941723008708`, 13.274605200446539`}, {3.5190689886612105`,
 12.314995185956427`}, {3.022101230720505`, 13.754410207691594`},
{2.5385650338052246`, 13.754410207691594`}, {5.1577194337629955`,
 18.47249277893464`}, {4.8353619691528085`, 17.193012759614493`},
{4.257804845059556`, 13.19463769923903`}, {3.9220158194239447`,
 15.433727733049288`}, {3.5593636717374837`, 15.353760231841779`},
{2.5922912779069223`, 10.79561266301375`}, {2.7669015712374403`,
 13.034702696824011`}, {4.15035235685616`, 16.71320775236944`},
```

```
{5.077130067610449`, 19.432102793424754`}, {4.78163572505111`,
 15.27379273063427`}, {3.7742686481442753`, 13.914345210106612`},
{3.39818493943239`, 13.994312711314121`}, {2.2565022522713107`,
 9.995937650938659`}, {2.8609224984154116`, 9.196262638863566`},
{4.566730748644319`, 16.3933377475394`}, {4.526436065568046`,
 14.15424771372914`}, {3.8011317701951244`, 13.274605200446539`},
{3.639953037890031`, 12.235027684748918`}, {3.1429852799493254`,
 12.474930188371445`}, {2.095323519966217`, 9.276230140071075`},
{2.2296391302204617`, 9.356197641278584`}, {4.1637839178815845`,
 13.674442706484085`}, {3.881721136347671`, 14.394150217351669`},
{3.6936792819917286`, 11.99512518112639`}, {3.2773008902035703`,
 12.394962687163936`}, {3.1698484020001745`, 13.754410207691594`},
{2.2296391302204617`, 9.516132643693602`}, {2.216207569195037`,
 8.956360135241038`}, {3.760837087118851`, 13.594475205276575`},
{3.7877002091697`, 12.394962687163936`}, {3.854858014296822`,
 10.715645161806242`}, {3.129553718923901`, 9.995937650938659`},
{2.7937646932882894`, 11.515320173881335`}, {2.269933813296735`,
 9.356197641278584`}, {2.122186642017066`, 7.197075108675834`},
{3.6936792819917286`, 12.714832691993974`}, {3.814563331220549`,
 10.075905152146168`}, {3.881721136347671`, 12.235027684748918`},
{3.129553718923901`, 10.635677660598732`}, {2.5788597168814977`,
 11.755222677503863`}, {2.0550288368899436`, 10.075905152146168`},
{2.2833653743221594`, 8.476555127995983`}, {3.881721136347671`,
 10.87558016422126`}, {4.02946830762734`, 11.195450169051298`},
{3.304164012254419`, 10.87558016422126`}, {3.3175955732798434`,
 12.874767694408993`}, {2.4848387897035265`, 11.915157679918881`},
{1.9878710317628212`, 8.556522629203492`}, {1.8804185435594254`,
 8.476555127995983`}, {2.901217181491685`, 9.995937650938659`},
{4.445846699415498`, 11.355385171466317`}, {4.15035235685616`,
 15.433727733049288`}, {4.136920795830736`, 13.434540202861557`},
{3.07582747482203`, 14.474117718559178`}, {1.7595344943306053`,
 8.156685123165945`}, {2.149049764067915`, 10.715645161806242`},
{3.733973965068002`, 14.07428021252163`}, {3.1698484020001745`,
 15.353760231841779`}, {2.7937646932882894`, 11.835190178711372`},
{2.7131753271357426`, 9.436165142486093`}, {2.498270350728951`,
 12.874767694408993`}, {2.296796935347584`, 7.996750120750927`},
{1.464040151771267`, 9.196262638863566`}, {3.8414264532713975`,
 13.034702696824011`}, {2.9952381086696565`, 10.475742658183714`},
{2.672880644059469`, 10.395775156976205`}, {2.4176809845764042`,
 10.475742658183714`}, {1.866986982534001`, 10.79561266301375`},
{1.7595344943306053`, 8.076717621958435`}, {1.5580610789492382`,
 7.197075108675834`}, {3.39818493943239`, 12.554897689578954`},
{3.0086696696950805`, 10.555710159391223`}, {2.4176809845764042`,
 8.076717621958435`}, {1.9610079097119724`, 8.076717621958435`},
```

```
{1.9744394707373967`, 7.277042609883344`}, {1.2625667363899`,
 9.036327636448547`}, {1.2088404922882021`, 6.077530091770704`},
{2.7400384491865912`, 7.516945113505871`}, {1.8804185435594254`,
 7.756847617128399`}, {2.0550288368899436`, 7.436977612298362`},
{2.01473415381367`, 8.956360135241038`}, {1.7729660553560298`,
 8.396587626788474`}, {1.3297245415170222`, 4.7180825712430465`},
{1.1013880040848063`, 7.357010111090853`}, {2.189344447144188`,
 9.596100144901111`}, {2.243070691245886`, 7.516945113505871`},
{1.7863976163814541`, 8.636490130411001`}, {1.5714926399746627`,
 6.157497592978213`}, {1.464040151771267`, 9.67606764610862`},
{1.0476617599831084`, 5.997562590563195`}, {0.9805039548559862`,
 5.277855079695612`}, {1.1013880040848063`, 7.197075108675834`},
{2.0550288368899436`, 9.356197641278584`}, {2.0281657148390946`,
 7.197075108675834`}, {1.719239811254332`, 6.157497592978213`},
{1.3431561025424468`, 6.477367597808251`}, {0.8999145887034393`,
 5.517757583318139`}, {0.9939355158814106`, 4.957985074865575`},
{1.840123860483152`, 6.957172605053306`}, {1.8804185435594254`,
 7.436977612298362`}, {1.611787323050936`, 6.55733509901576`}};
```

Enter the following to avoid upload:

```
In[∘]:= dates = tabdates;
```

```
In[∘]:= For[i = 1, i <= Length[dates], i++, datetabQC[i] = tab[[i]]]
```

Overall Analysis by 70+

Cubic Spline fit (https://mathematica.stackexchange.com/questions/33206/implementation-of-smoothing-splines-function)

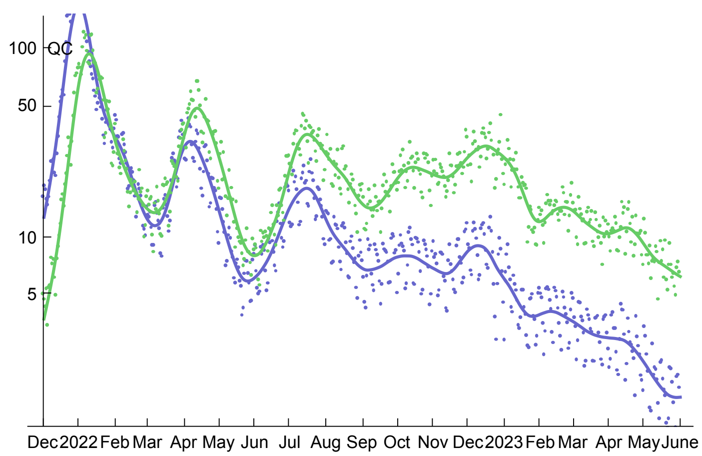
```
In[∘]:= CubicSplSmooth[data_, lambda_] := Module[{M, Knots, X, Dsq, a}, M = Length@data;
    Knots = Flatten@{1, 1, 1, Range@M, M, M, M};
    X = Table[Evaluate@N@BSplineBasis[{3, Knots}, n, t], {t, 1, M}, {n, 0, M + 1}];
    Dsq = Differences[X, 2];
    a = LinearSolve[Transpose[X].X + lambda * Transpose[Dsq].Dsq,
      Transpose[X].data, Method → "Multifrontal"];
    Return[
     X.
      a]];
```

Cubic spline fit to all dates (Y is younger, A is 70+), using lambda = 3 to provide a good fit but not overfit to the data:

```
In[∘]:= modYQC = CubicSplSmooth[Table[{i, Log[Max[1, datetabQC[i][[1]]]]},
      {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
    modAQC = CubicSplSmooth[Table[{i, Log[Max[1, datetabQC[i][[2]]]]},
      {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
```

In[∘]:=
```
plotQC = Show[
  ListPlot[Table[{i, Log[datetabQC[i][[1]]]}, {i, 1, Length[dates]}],
   PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.4, 0.8]}],
  ListLinePlot[modYQC, PlotStyle → RGBColor[0.4, 0.4, 0.8], PlotRange → All],
  ListLinePlot[modAQC, PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
  ListPlot[Table[{i, Log[datetabQC[i][[2]]]}, {i, 1, Length[dates]}],
   PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
  Graphics[Text["QC", {5, 4.6}, {-1, 0}]],
  PlotRange → {Automatic, {0, 5}}, AxesOrigin → {1, 0},
  Ticks → {xaxis, {{Log[5], "5"}, {Log[10], "10"}, {Log[50], "50"}, {Log[100], "100"},
     {Log[500], "500"}, {Log[1000], "1000"}, {Log[5000], "5000"}}}
 ]
```

Out[∘]=

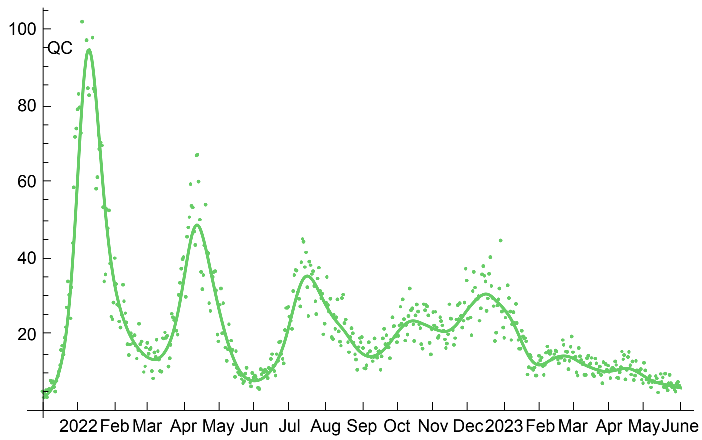


The above illustrates the major drop off in reported cases among those <70:

70+ on a non-log plot

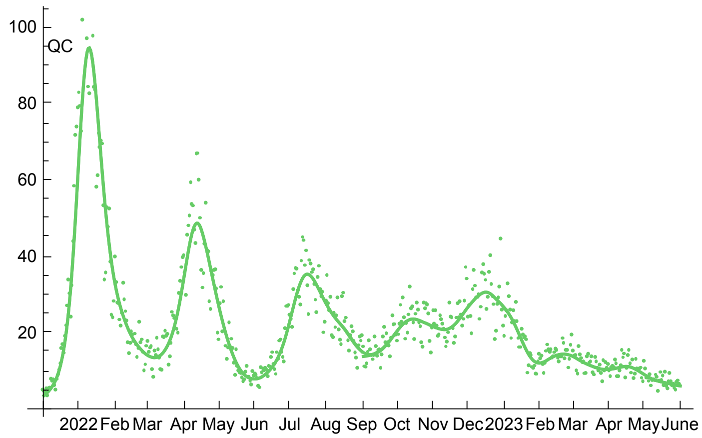
In[◦]:=
```
Show[
 ListLinePlot[Table[{modAQC[[i, 1]], Exp[modAQC[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabQC[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["QC", {5, 95}, {-1, 0}]],
 PlotRange → {Automatic, {0, 100}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

Out[◦]=



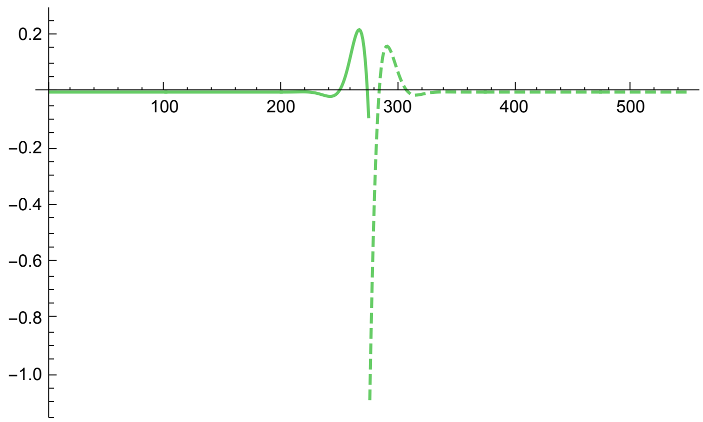
In[◦]:=
```
modAQC1 = CubicSplSmooth[Table[{i, Log[Max[1, datetabQC[i][[2]]]]},
    {i, 1, Position[dates, myenddate1][[1, 1]]}], 10^lambda /. lambda → 3];
modAQC2 = CubicSplSmooth[Table[{i, Log[Max[1, datetabQC[i][[2]]]]},
    {i, Position[dates, mystartdate2][[1, 1]], Length[dates]}],
   10^lambda /. lambda → 3];
```

In[•]:=
```
Show[
 ListLinePlot[Table[{modAQC[[i, 1]], Exp[modAQC[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAQC1[[i, 1]], Exp[modAQC1[[i, 2]]]},
   {i, 1, Length[modAQC1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAQC2[[i, 1]], Exp[modAQC2[[i, 2]]]},
   {i, 1, Length[modAQC2]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabQC[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["QC", {5, 95}, {-1, 0}]],
 PlotRange → {Automatic, {0, 100}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

Out[•]=


The fit is very similar, with a slight discrepancy in numbers at the joint, so we just use the overall spline fit for the analysis:

In[∘]:=
```
Show[
 ListLinePlot[Table[{modAQC1[[i, 1]], Exp[modAQC1[[i, 2]]] - Exp[modAQC[[i, 2]]]},
   {i, 1, Length[modAQC1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAQC2[[i, 1]], Exp[modAQC2[[i, 2]]] - Exp[modAQC[[
      i + Position[dates, myenddate1][[1, 1]], 2]]]}, {i, 1, Length[modAQC2]}],
  PlotStyle → {Dashed, RGBColor[0.4, 0.8, 0.4]}, PlotRange → All],
 PlotRange → All
]
```

Out[∘]=


Merging with Omicron subvariant frequencies

In[∘]:=
```
graphheight = 120;
```

In[∘]:=
```
colourBA1 = RGBColor[0.5, 0.1, 0.1];
colourBA11 = RGBColor[0.75, 0.1, 0.1];
colourBA2 = RGBColor[1, 0.1, 0.1];
colourBA5 = RGBColor[1, 0.5, 0.1];
colourBA45 = colourBA5;
colourBA4 = colourBA5;
colourBQ = RGBColor[1, 0.75, 0.5];
colourXBB15 = RGBColor[0.2, 0.5, 0.8];
colourXBBO = RGBColor[0.5, 0.7, 0.8];

{colourBA1, colourBA11, colourBA2, colourBA45, colourBQ, colourXBB15, colourXBBO}
```

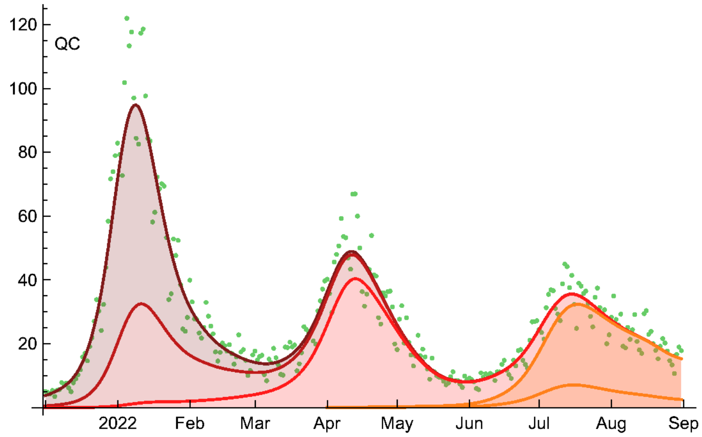
Out[∘]=


From Omicron_VariantsGISAID.nb (selection on BA.1, BA.1.1 and BA.2 calculated from 16 May analysis; selection on BA.4&BA.5 from 24 June analysis where the initial frequency is for “2022-04-1” at t=1):

In[∘]:=
```
QCML = {pBA11 → 0.1299249754084715`, pBA2 → 0.000681917154561101`,
   sBA11 → 0.029470405369802218`, sBA2 → 0.0814267906019519`,
   pBA4 → 0.00023463957846716374`, pBA5 → 0.00036479142777358456`,
   sBA4 → 0.0853074671375297`, sBA5 → 0.09303958537855452`};
```

```
In[∘]:= endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modAQC[[i, 1]],
    1 * Exp[modAQC[[i, 2]]] /. t -> modAQC[[i, 1]] /. QCML}, {i, 1, endtime}];
tabBA11 = Table[{modAQC[[i, 1]],
    (pBA2 Exp[sBA2 t] + pBA11 Exp[sBA11 t]) / ((1 - pBA11 - pBA2) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t]) * Exp[modAQC[[i, 2]]] /.
      t -> modAQC[[i, 1]] /. QCML}, {i, 1, endtime}];
tabBA2 = Table[{modAQC[[i, 1]],
    (pBA2 Exp[sBA2 t]) / ((1 - pBA11 - pBA2) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t]) * Exp[modAQC[[i, 2]]] /.
      t -> modAQC[[i, 1]] /. QCML}, {i, 1, endtime}];
tabBA4 = Table[{modAQC[[i, 1]], (pBA4 Exp[sBA4 t]) / ((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
       Exp[modAQC[[i, 2]]] /. t -> modAQC[[i, 1]] - curdate["2022-03-31"] /.
     QCML}, {i, curdate["2022-04-1"], endtime}];
tabBA5 = Table[{modAQC[[i, 1]], (pBA5 Exp[sBA5 t] + pBA4 Exp[sBA4 t]) / ((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
       Exp[modAQC[[i, 2]]] /. t -> modAQC[[i, 1]] - curdate["2022-03-31"] /.
     QCML}, {i, curdate["2022-04-1"], endtime}];

Show[
 ListLinePlot[Table[{modAQC[[i, 1]], Exp[modAQC[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabQC[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA4, tabBA5}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
     Filling → {1 → {2}, 2 → {3}, 3 → Axis, 4 → Axis, 5 → {4}}],
 Graphics[Text["QC", {5, 0.95 graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[∘]=



From duotang/VirusSeq (see code above): mutantNames = list(“BA.1.1*”, “BA.2*”, “BA.4/5*”, “BA.1*”)

In[∘]:=
```
QCML1
```

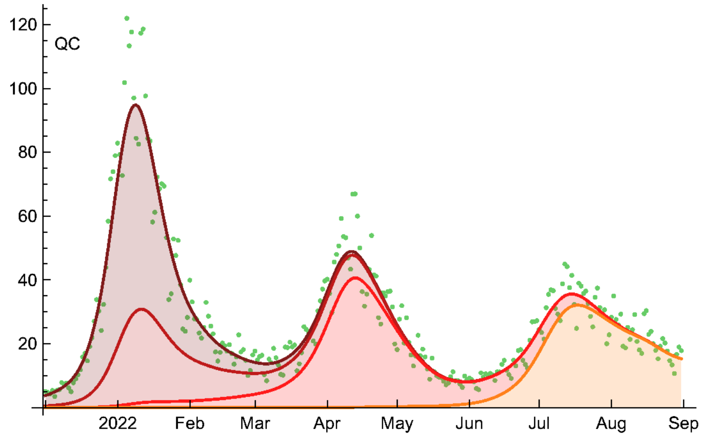
Out[∘]=
```
{0.130482, 0.848161, 0.00250028, 0.0285217, 0.0800771, 0.167335}
```

In[∘]:=
```
dataname = QCML1;
refdate = "2022-04-15"; (* t=0 date used in duotang code above *)
QCML = {pBA11 → dataname[[1]], pBA2 → dataname[[2]], pBA45 → dataname[[3]],
  sBA11 → dataname[[4]], sBA2 → dataname[[5]], sBA45 → dataname[[6]]}
```

Out[∘]=
```
{pBA11 → 0.130482, pBA2 → 0.848161, pBA45 → 0.00250028,
 sBA11 → 0.0285217, sBA2 → 0.0800771, sBA45 → 0.167335}
```

In[∘]:=
```
endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modAQC[[i, 1]],
    1 * Exp[modAQC[[i, 2]]] /. t -> modAQC[[i, 1]] /. QCML}, {i, 1, endtime}];
tabBA11 = Table[{modAQC[[i, 1]],
    (pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAQC[[i, 2]]] /.
     t -> modAQC[[i, 1]] - curdate[refdate] /. QCML}, {i, 1, endtime}];
tabBA2 = Table[{modAQC[[i, 1]],
    (pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAQC[[i, 2]]] /.
     t -> modAQC[[i, 1]] - curdate[refdate] /. QCML}, {i, 1, endtime}];
tabBA45 = Table[{modAQC[[i, 1]],
    (pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAQC[[i, 2]]] /.
     t -> modAQC[[i, 1]] - curdate[refdate] /. QCML}, {i, 1, endtime}];

Show[
 ListLinePlot[Table[{modAQC[[i, 1]], Exp[modAQC[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabQC[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
      Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis}],
 Graphics[Text["QC", {5, 0.95 graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[∘]=



From duotang/VirusSeq (see code above): mutantNames = list(“XBB.1.5*”, “XBB.Other*”, “BQ*”, “BA.5*”)

In[∘]:=
```
QCML2
```

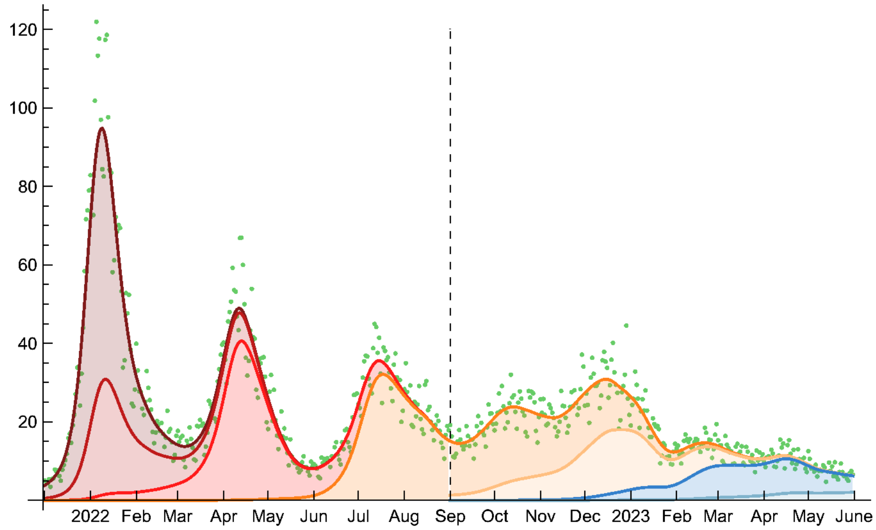
Out[∘]=
```
{0.170115, 0.00588832, 0.579925, 0.0636735, 0.0828669, 0.0237981}
```

In[∘]:=
```
dataname = QCML2;
refdate = "2023-01-14"; (* t=0 date used in duotang code above *)
QCML = {pXBB15 → dataname[[1]], pXBBO → dataname[[2]], pBQ → dataname[[3]],
   sXBB15 → dataname[[4]], sXBBO → dataname[[5]], sBQ → dataname[[6]]}
```

Out[∘]=
```
{pXBB15 → 0.170115, pXBBO → 0.00588832, pBQ → 0.579925,
 sXBB15 → 0.0636735, sXBBO → 0.0828669, sBQ → 0.0237981}
```

In[∘]:=
```
starttime = Position[dates, "2022-09-01"][[1, 1]];
endtime = Length[dates];

tabBA5 = Table[{modAQC[[i, 1]], 1 * Exp[modAQC[[i, 2]]] /. t -> modAQC[[i, 1]] /. QCML},
   {i, starttime, endtime}];
tabBQ = Table[{modAQC[[i, 1]],
     (pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
     ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
       Exp[modAQC[[i, 2]]] /.
      t -> modAQC[[i, 1]] - curdate[refdate] /. QCML}, {i, starttime, endtime}];
tabXBB15 = Table[{modAQC[[i, 1]],
     (pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
     ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
       Exp[modAQC[[i, 2]]] /.
      t -> modAQC[[i, 1]] - curdate[refdate] /. QCML}, {i, starttime, endtime}];
tabXBBO = Table[{modAQC[[i, 1]],
     (pXBBO Exp[sXBBO t]) /
     ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
       Exp[modAQC[[i, 2]]] /.
      t -> modAQC[[i, 1]] - curdate[refdate] /. QCML}, {i, starttime, endtime}];

Show[
 ListLinePlot[Table[{modAQC[[i, 1]], Exp[modAQC[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabQC[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45, tabBA5, tabBQ, tabXBB15, tabXBBO},
  PlotStyle → {colourBA1, colourBA11, colourBA2, colourBA45,
    colourBA45, colourBQ, colourXBB15, colourXBBO}, PlotRange → All,
      Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis,
    5 → {6}, 6 → {7}, 7 → {8}, 8 → Axis}],
 (*Graphics[Text["QC",{5,95},{-1,0}]],*)
 ListPlot[{{starttime, 0}, {starttime, 120}},
  Joined → True, PlotStyle → {Black, Thickness[0.002], Dashed}],
 PlotRange → {{1, curdate["2023-06-01"]}, {0, graphheight}},
 Ticks → {xaxis, Automatic},
 AspectRatio → 0.6,
 ImageSize → 450
]
```

Out[∘]=



ON - 70+ case counts and genomic frequencies (Dec 2021-May 2023)

Using “Accurate_Episode_Date” and correcting recent dates from the dashboard for ON-wide:

https://data.ontario.ca/dataset/confirmed-positive-cases-of-covid-19-in-ontario/resource/455fd63b-603d-4608-8216-7d8647f43350

Processing ON Case Data [ENTER this or the next folder - uploading & processing data]

In[∘]:=
```
SetDirectory["/Users/otto/Dropbox/COVID19/BC_Data/CorrectingCaseCounts"];
```

In[∘]:=
```
SetDirectory["/Users/otto/Downloads"];
```

In[∘]:=
```
startafter = "2021-11-30";
```

The two periods we’ll consider:

In[∘]:=
```
mystartdate1 = "2021-12-01";
myenddate1 = "2022-08-31";
mystartdate2 = "2022-09-01";
myenddate2 = "2023-05-31";
```

In[∘]:=
```
curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

In[◦]:=
```
xaxis = {{curdate["2021-10-01"], "Oct"},
   {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
   {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
   {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
   {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
   {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
   {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
   {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
   {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
   {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
   {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

Uploading the data (see below to avoid this step):

In[◦]:=
```
temp0 = Import["conposcovidloc16Jun2023.csv"];
temp0[[1]]
Drop[temp0, 1];
temp = %[[588000 ;; Length[%]]]; (*Limiting data to last three months*)
counts = Transpose[{temp[[All, 2]], temp[[All, 12]], temp[[All, 6]]}];
Clear[temp0, temp]
```

Out[◦]=
```
{Row_ID, Accurate_Episode_Date, Case_Reported_Date, Test_Reported_Date,
 Specimen_Date, Age_Group, Client_Gender, Outcome1, Reporting_PHU_ID, Reporting_PHU,
 Reporting_PHU_Address, Reporting_PHU_City, Reporting_PHU_Postal_Code,
 Reporting_PHU_Website, Reporting_PHU_Latitude, Reporting_PHU_Longitude}
```

In[◦]:=
```
Last[Position[counts, startafter]][[1]]
```

Out[◦]= 35826

In[◦]:=
```
counts = counts[[% + 1 ;; Length[counts]]];
```

In[◦]:=
```
dates = DeleteDuplicates[counts[[All, 1]]]
```

Out[◦]=
```
{2021-12-01, 2021-12-02, 2021-12-03, 2021-12-04, 2021-12-05, 2021-12-06, 2021-12-07,
 2021-12-08, 2021-12-09, 2021-12-10, 2021-12-11, 2021-12-12, 2021-12-13, 2021-12-14,
 2021-12-15, 2021-12-16, 2021-12-17, 2021-12-18, 2021-12-19, 2021-12-20,
 2021-12-21, 2021-12-22, 2021-12-23, 2021-12-24, 2021-12-25, 2021-12-26,
 2021-12-27, 2021-12-28, 2021-12-29, 2021-12-30, 2021-12-31, 2022-01-01,
 2022-01-02, 2022-01-03, 2022-01-04, 2022-01-05, 2022-01-06, 2022-01-07,
 2022-01-08, 2022-01-09, 2022-01-10, 2022-01-11, 2022-01-12, 2022-01-13,
 2022-01-14, 2022-01-15, 2022-01-16, 2022-01-17, 2022-01-18, 2022-01-19,
 2022-01-20, 2022-01-21, 2022-01-22, 2022-01-23, 2022-01-24, 2022-01-25,
 2022-01-26, 2022-01-27, 2022-01-28, 2022-01-29, 2022-01-30, 2022-01-31,
 2022-02-01, 2022-02-02, 2022-02-03, 2022-02-04, 2022-02-05, 2022-02-06,
 2022-02-07, 2022-02-08, 2022-02-09, 2022-02-10, 2022-02-11, 2022-02-12,
 2022-02-13, 2022-02-14, 2022-02-15, 2022-02-16, 2022-02-17, 2022-02-18,
 2022-02-19, 2022-02-20, 2022-02-21, 2022-02-22, 2022-02-23, 2022-02-24,
```

```
2022-02-25, 2022-02-26, 2022-02-27, 2022-02-28, 2022-03-01, 2022-03-02,
2022-03-03, 2022-03-04, 2022-03-05, 2022-03-06, 2022-03-07, 2022-03-08,
2022-03-09, 2022-03-10, 2022-03-11, 2022-03-12, 2022-03-13, 2022-03-14,
2022-03-15, 2022-03-16, 2022-03-17, 2022-03-18, 2022-03-19, 2022-03-20,
2022-03-21, 2022-03-22, 2022-03-23, 2022-03-24, 2022-03-25, 2022-03-26,
2022-03-27, 2022-03-28, 2022-03-29, 2022-03-30, 2022-03-31, 2022-04-01,
2022-04-02, 2022-04-03, 2022-04-04, 2022-04-05, 2022-04-06, 2022-04-07,
2022-04-08, 2022-04-09, 2022-04-10, 2022-04-11, 2022-04-12, 2022-04-13,
2022-04-14, 2022-04-15, 2022-04-16, 2022-04-17, 2022-04-18, 2022-04-19,
2022-04-20, 2022-04-21, 2022-04-22, 2022-04-23, 2022-04-24, 2022-04-25,
2022-04-26, 2022-04-27, 2022-04-28, 2022-04-29, 2022-04-30, 2022-05-01,
2022-05-02, 2022-05-03, 2022-05-04, 2022-05-05, 2022-05-06, 2022-05-07,
2022-05-08, 2022-05-09, 2022-05-10, 2022-05-11, 2022-05-12, 2022-05-13,
2022-05-14, 2022-05-15, 2022-05-16, 2022-05-17, 2022-05-18, 2022-05-19,
2022-05-20, 2022-05-21, 2022-05-22, 2022-05-23, 2022-05-24, 2022-05-25,
2022-05-26, 2022-05-27, 2022-05-28, 2022-05-29, 2022-05-30, 2022-05-31,
2022-06-01, 2022-06-02, 2022-06-03, 2022-06-04, 2022-06-05, 2022-06-06,
2022-06-07, 2022-06-08, 2022-06-09, 2022-06-10, 2022-06-11, 2022-06-12,
2022-06-13, 2022-06-14, 2022-06-15, 2022-06-16, 2022-06-17, 2022-06-18,
2022-06-19, 2022-06-20, 2022-06-21, 2022-06-22, 2022-06-23, 2022-06-24,
2022-06-25, 2022-06-26, 2022-06-27, 2022-06-28, 2022-06-29, 2022-06-30,
2022-07-01, 2022-07-02, 2022-07-03, 2022-07-04, 2022-07-05, 2022-07-06,
2022-07-07, 2022-07-08, 2022-07-09, 2022-07-10, 2022-07-11, 2022-07-12,
2022-07-13, 2022-07-14, 2022-07-15, 2022-07-16, 2022-07-17, 2022-07-18,
2022-07-19, 2022-07-20, 2022-07-21, 2022-07-22, 2022-07-23, 2022-07-24,
2022-07-25, 2022-07-26, 2022-07-27, 2022-07-28, 2022-07-29, 2022-07-30,
2022-07-31, 2022-08-01, 2022-08-02, 2022-08-03, 2022-08-04, 2022-08-05,
2022-08-06, 2022-08-07, 2022-08-08, 2022-08-09, 2022-08-10, 2022-08-11,
2022-08-12, 2022-08-13, 2022-08-14, 2022-08-15, 2022-08-16, 2022-08-17,
2022-08-18, 2022-08-19, 2022-08-20, 2022-08-21, 2022-08-22, 2022-08-23,
2022-08-24, 2022-08-25, 2022-08-26, 2022-08-27, 2022-08-28, 2022-08-29,
2022-08-30, 2022-08-31, 2022-09-01, 2022-09-02, 2022-09-03, 2022-09-04,
2022-09-05, 2022-09-06, 2022-09-07, 2022-09-08, 2022-09-09, 2022-09-10,
2022-09-11, 2022-09-12, 2022-09-13, 2022-09-14, 2022-09-15, 2022-09-16,
2022-09-17, 2022-09-18, 2022-09-19, 2022-09-20, 2022-09-21, 2022-09-22,
2022-09-23, 2022-09-24, 2022-09-25, 2022-09-26, 2022-09-27, 2022-09-28,
2022-09-29, 2022-09-30, 2022-10-01, 2022-10-02, 2022-10-03, 2022-10-04,
2022-10-05, 2022-10-06, 2022-10-07, 2022-10-08, 2022-10-09, 2022-10-10,
2022-10-11, 2022-10-12, 2022-10-13, 2022-10-14, 2022-10-15, 2022-10-16,
2022-10-17, 2022-10-18, 2022-10-19, 2022-10-20, 2022-10-21, 2022-10-22,
2022-10-23, 2022-10-24, 2022-10-25, 2022-10-26, 2022-10-27, 2022-10-28,
2022-10-29, 2022-10-30, 2022-10-31, 2022-11-01, 2022-11-02, 2022-11-03,
2022-11-04, 2022-11-05, 2022-11-06, 2022-11-07, 2022-11-08, 2022-11-09,
2022-11-10, 2022-11-11, 2022-11-12, 2022-11-13, 2022-11-14, 2022-11-15,
```

```
2022-11-16, 2022-11-17, 2022-11-18, 2022-11-19, 2022-11-20, 2022-11-21,
2022-11-22, 2022-11-23, 2022-11-24, 2022-11-25, 2022-11-26, 2022-11-27,
2022-11-28, 2022-11-29, 2022-11-30, 2022-12-01, 2022-12-02, 2022-12-03,
2022-12-04, 2022-12-05, 2022-12-06, 2022-12-07, 2022-12-08, 2022-12-09,
2022-12-10, 2022-12-11, 2022-12-12, 2022-12-13, 2022-12-14, 2022-12-15,
2022-12-16, 2022-12-17, 2022-12-18, 2022-12-19, 2022-12-20, 2022-12-21,
2022-12-22, 2022-12-23, 2022-12-24, 2022-12-25, 2022-12-26, 2022-12-27,
2022-12-28, 2022-12-29, 2022-12-30, 2022-12-31, 2023-01-01, 2023-01-02,
2023-01-03, 2023-01-04, 2023-01-05, 2023-01-06, 2023-01-07, 2023-01-08,
2023-01-09, 2023-01-10, 2023-01-11, 2023-01-12, 2023-01-13, 2023-01-14,
2023-01-15, 2023-01-16, 2023-01-17, 2023-01-18, 2023-01-19, 2023-01-20,
2023-01-21, 2023-01-22, 2023-01-23, 2023-01-24, 2023-01-25, 2023-01-26,
2023-01-27, 2023-01-28, 2023-01-29, 2023-01-30, 2023-01-31, 2023-02-01,
2023-02-02, 2023-02-03, 2023-02-04, 2023-02-05, 2023-02-06, 2023-02-07,
2023-02-08, 2023-02-09, 2023-02-10, 2023-02-11, 2023-02-12, 2023-02-13,
2023-02-14, 2023-02-15, 2023-02-16, 2023-02-17, 2023-02-18, 2023-02-19,
2023-02-20, 2023-02-21, 2023-02-22, 2023-02-23, 2023-02-24, 2023-02-25,
2023-02-26, 2023-02-27, 2023-02-28, 2023-03-01, 2023-03-02, 2023-03-03,
2023-03-04, 2023-03-05, 2023-03-06, 2023-03-07, 2023-03-08, 2023-03-09,
2023-03-10, 2023-03-11, 2023-03-12, 2023-03-13, 2023-03-14, 2023-03-15,
2023-03-16, 2023-03-17, 2023-03-18, 2023-03-19, 2023-03-20, 2023-03-21,
2023-03-22, 2023-03-23, 2023-03-24, 2023-03-25, 2023-03-26, 2023-03-27,
2023-03-28, 2023-03-29, 2023-03-30, 2023-03-31, 2023-04-01, 2023-04-02,
2023-04-03, 2023-04-04, 2023-04-05, 2023-04-06, 2023-04-07, 2023-04-08,
2023-04-09, 2023-04-10, 2023-04-11, 2023-04-12, 2023-04-13, 2023-04-14,
2023-04-15, 2023-04-16, 2023-04-17, 2023-04-18, 2023-04-19, 2023-04-20,
2023-04-21, 2023-04-22, 2023-04-23, 2023-04-24, 2023-04-25, 2023-04-26,
2023-04-27, 2023-04-28, 2023-04-29, 2023-04-30, 2023-05-01, 2023-05-02,
2023-05-03, 2023-05-04, 2023-05-05, 2023-05-06, 2023-05-07, 2023-05-08,
2023-05-09, 2023-05-10, 2023-05-11, 2023-05-12, 2023-05-13, 2023-05-14,
2023-05-15, 2023-05-16, 2023-05-17, 2023-05-18, 2023-05-19, 2023-05-20,
2023-05-21, 2023-05-22, 2023-05-23, 2023-05-24, 2023-05-25, 2023-05-26,
2023-05-27, 2023-05-28, 2023-05-29, 2023-05-30, 2023-05-31, 2023-06-01,
2023-06-02, 2023-06-03, 2023-06-04, 2023-06-05, 2023-06-06, 2023-06-07,
2023-06-08, 2023-06-09, 2023-06-10, 2023-06-11, 2023-06-12, 2023-06-13}
```

Age list (UNKNOWN treated as <70):

```
In[]:= DeleteDuplicates[counts[[All, 3]]]
Out[]= {50s, <20, 20s, 30s, 40s, 60s, 70s, 80s, 90+, UNKNOWN}
```

Trim off data from Quebec after end date (this data stream keeps going until June 13, 2023):

```
In[]:= Last[Position[counts, myenddate2]][[1]]
Out[]= 998 564
```

```
In[•]:= counts = counts[[1 ;; %]];

In[•]:= dates = DeleteDuplicates[counts[[All, 1]]]

Out[•]= {2021-12-01, 2021-12-02, 2021-12-03, 2021-12-04, 2021-12-05, 2021-12-06, 2021-12-07,
  2021-12-08, 2021-12-09, 2021-12-10, 2021-12-11, 2021-12-12, 2021-12-13, 2021-12-14,
  2021-12-15, 2021-12-16, 2021-12-17, 2021-12-18, 2021-12-19, 2021-12-20, 2021-12-21,
  2021-12-22, 2021-12-23, 2021-12-24, 2021-12-25, 2021-12-26, 2021-12-27, 2021-12-28,
  2021-12-29, 2021-12-30, 2021-12-31, 2022-01-01, 2022-01-02, 2022-01-03,
  2022-01-04, 2022-01-05, 2022-01-06, 2022-01-07, 2022-01-08, 2022-01-09,
  2022-01-10, 2022-01-11, 2022-01-12, 2022-01-13, 2022-01-14, 2022-01-15,
  2022-01-16, 2022-01-17, 2022-01-18, 2022-01-19, 2022-01-20, 2022-01-21,
  2022-01-22, 2022-01-23, 2022-01-24, 2022-01-25, 2022-01-26, 2022-01-27,
  2022-01-28, 2022-01-29, 2022-01-30, 2022-01-31, 2022-02-01, 2022-02-02,
  2022-02-03, 2022-02-04, 2022-02-05, 2022-02-06, 2022-02-07, 2022-02-08,
  2022-02-09, 2022-02-10, 2022-02-11, 2022-02-12, 2022-02-13, 2022-02-14,
  2022-02-15, 2022-02-16, 2022-02-17, 2022-02-18, 2022-02-19, 2022-02-20,
  2022-02-21, 2022-02-22, 2022-02-23, 2022-02-24, 2022-02-25, 2022-02-26,
  2022-02-27, 2022-02-28, 2022-03-01, 2022-03-02, 2022-03-03, 2022-03-04,
  2022-03-05, 2022-03-06, 2022-03-07, 2022-03-08, 2022-03-09, 2022-03-10,
  2022-03-11, 2022-03-12, 2022-03-13, 2022-03-14, 2022-03-15, 2022-03-16,
  2022-03-17, 2022-03-18, 2022-03-19, 2022-03-20, 2022-03-21, 2022-03-22,
  2022-03-23, 2022-03-24, 2022-03-25, 2022-03-26, 2022-03-27, 2022-03-28,
  2022-03-29, 2022-03-30, 2022-03-31, 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04,
  2022-04-05, 2022-04-06, 2022-04-07, 2022-04-08, 2022-04-09, 2022-04-10,
  2022-04-11, 2022-04-12, 2022-04-13, 2022-04-14, 2022-04-15, 2022-04-16,
  2022-04-17, 2022-04-18, 2022-04-19, 2022-04-20, 2022-04-21, 2022-04-22,
  2022-04-23, 2022-04-24, 2022-04-25, 2022-04-26, 2022-04-27, 2022-04-28,
  2022-04-29, 2022-04-30, 2022-05-01, 2022-05-02, 2022-05-03, 2022-05-04,
  2022-05-05, 2022-05-06, 2022-05-07, 2022-05-08, 2022-05-09, 2022-05-10,
  2022-05-11, 2022-05-12, 2022-05-13, 2022-05-14, 2022-05-15, 2022-05-16,
  2022-05-17, 2022-05-18, 2022-05-19, 2022-05-20, 2022-05-21, 2022-05-22,
  2022-05-23, 2022-05-24, 2022-05-25, 2022-05-26, 2022-05-27, 2022-05-28,
  2022-05-29, 2022-05-30, 2022-05-31, 2022-06-01, 2022-06-02, 2022-06-03,
  2022-06-04, 2022-06-05, 2022-06-06, 2022-06-07, 2022-06-08, 2022-06-09,
  2022-06-10, 2022-06-11, 2022-06-12, 2022-06-13, 2022-06-14, 2022-06-15,
  2022-06-16, 2022-06-17, 2022-06-18, 2022-06-19, 2022-06-20, 2022-06-21,
  2022-06-22, 2022-06-23, 2022-06-24, 2022-06-25, 2022-06-26, 2022-06-27,
  2022-06-28, 2022-06-29, 2022-06-30, 2022-07-01, 2022-07-02, 2022-07-03, 2022-07-04,
  2022-07-05, 2022-07-06, 2022-07-07, 2022-07-08, 2022-07-09, 2022-07-10,
  2022-07-11, 2022-07-12, 2022-07-13, 2022-07-14, 2022-07-15, 2022-07-16,
  2022-07-17, 2022-07-18, 2022-07-19, 2022-07-20, 2022-07-21, 2022-07-22,
  2022-07-23, 2022-07-24, 2022-07-25, 2022-07-26, 2022-07-27, 2022-07-28,
  2022-07-29, 2022-07-30, 2022-07-31, 2022-08-01, 2022-08-02, 2022-08-03,
  2022-08-04, 2022-08-05, 2022-08-06, 2022-08-07, 2022-08-08, 2022-08-09,
```

```
2022-08-10, 2022-08-11, 2022-08-12, 2022-08-13, 2022-08-14, 2022-08-15,
2022-08-16, 2022-08-17, 2022-08-18, 2022-08-19, 2022-08-20, 2022-08-21,
2022-08-22, 2022-08-23, 2022-08-24, 2022-08-25, 2022-08-26, 2022-08-27,
2022-08-28, 2022-08-29, 2022-08-30, 2022-08-31, 2022-09-01, 2022-09-02,
2022-09-03, 2022-09-04, 2022-09-05, 2022-09-06, 2022-09-07, 2022-09-08,
2022-09-09, 2022-09-10, 2022-09-11, 2022-09-12, 2022-09-13, 2022-09-14,
2022-09-15, 2022-09-16, 2022-09-17, 2022-09-18, 2022-09-19, 2022-09-20,
2022-09-21, 2022-09-22, 2022-09-23, 2022-09-24, 2022-09-25, 2022-09-26, 2022-09-27,
2022-09-28, 2022-09-29, 2022-09-30, 2022-10-01, 2022-10-02, 2022-10-03,
2022-10-04, 2022-10-05, 2022-10-06, 2022-10-07, 2022-10-08, 2022-10-09,
2022-10-10, 2022-10-11, 2022-10-12, 2022-10-13, 2022-10-14, 2022-10-15,
2022-10-16, 2022-10-17, 2022-10-18, 2022-10-19, 2022-10-20, 2022-10-21,
2022-10-22, 2022-10-23, 2022-10-24, 2022-10-25, 2022-10-26, 2022-10-27,
2022-10-28, 2022-10-29, 2022-10-30, 2022-10-31, 2022-11-01, 2022-11-02,
2022-11-03, 2022-11-04, 2022-11-05, 2022-11-06, 2022-11-07, 2022-11-08,
2022-11-09, 2022-11-10, 2022-11-11, 2022-11-12, 2022-11-13, 2022-11-14,
2022-11-15, 2022-11-16, 2022-11-17, 2022-11-18, 2022-11-19, 2022-11-20,
2022-11-21, 2022-11-22, 2022-11-23, 2022-11-24, 2022-11-25, 2022-11-26,
2022-11-27, 2022-11-28, 2022-11-29, 2022-11-30, 2022-12-01, 2022-12-02,
2022-12-03, 2022-12-04, 2022-12-05, 2022-12-06, 2022-12-07, 2022-12-08,
2022-12-09, 2022-12-10, 2022-12-11, 2022-12-12, 2022-12-13, 2022-12-14,
2022-12-15, 2022-12-16, 2022-12-17, 2022-12-18, 2022-12-19, 2022-12-20,
2022-12-21, 2022-12-22, 2022-12-23, 2022-12-24, 2022-12-25, 2022-12-26,
2022-12-27, 2022-12-28, 2022-12-29, 2022-12-30, 2022-12-31, 2023-01-01,
2023-01-02, 2023-01-03, 2023-01-04, 2023-01-05, 2023-01-06, 2023-01-07,
2023-01-08, 2023-01-09, 2023-01-10, 2023-01-11, 2023-01-12, 2023-01-13,
2023-01-14, 2023-01-15, 2023-01-16, 2023-01-17, 2023-01-18, 2023-01-19,
2023-01-20, 2023-01-21, 2023-01-22, 2023-01-23, 2023-01-24, 2023-01-25,
2023-01-26, 2023-01-27, 2023-01-28, 2023-01-29, 2023-01-30, 2023-01-31,
2023-02-01, 2023-02-02, 2023-02-03, 2023-02-04, 2023-02-05, 2023-02-06,
2023-02-07, 2023-02-08, 2023-02-09, 2023-02-10, 2023-02-11, 2023-02-12,
2023-02-13, 2023-02-14, 2023-02-15, 2023-02-16, 2023-02-17, 2023-02-18,
2023-02-19, 2023-02-20, 2023-02-21, 2023-02-22, 2023-02-23, 2023-02-24,
2023-02-25, 2023-02-26, 2023-02-27, 2023-02-28, 2023-03-01, 2023-03-02,
2023-03-03, 2023-03-04, 2023-03-05, 2023-03-06, 2023-03-07, 2023-03-08,
2023-03-09, 2023-03-10, 2023-03-11, 2023-03-12, 2023-03-13, 2023-03-14,
2023-03-15, 2023-03-16, 2023-03-17, 2023-03-18, 2023-03-19, 2023-03-20,
2023-03-21, 2023-03-22, 2023-03-23, 2023-03-24, 2023-03-25, 2023-03-26,
2023-03-27, 2023-03-28, 2023-03-29, 2023-03-30, 2023-03-31, 2023-04-01,
2023-04-02, 2023-04-03, 2023-04-04, 2023-04-05, 2023-04-06, 2023-04-07,
2023-04-08, 2023-04-09, 2023-04-10, 2023-04-11, 2023-04-12, 2023-04-13,
2023-04-14, 2023-04-15, 2023-04-16, 2023-04-17, 2023-04-18, 2023-04-19,
2023-04-20, 2023-04-21, 2023-04-22, 2023-04-23, 2023-04-24, 2023-04-25,
2023-04-26, 2023-04-27, 2023-04-28, 2023-04-29, 2023-04-30, 2023-05-01,
```

```
2023-05-02, 2023-05-03, 2023-05-04, 2023-05-05, 2023-05-06, 2023-05-07,
2023-05-08, 2023-05-09, 2023-05-10, 2023-05-11, 2023-05-12, 2023-05-13,
2023-05-14, 2023-05-15, 2023-05-16, 2023-05-17, 2023-05-18, 2023-05-19,
2023-05-20, 2023-05-21, 2023-05-22, 2023-05-23, 2023-05-24, 2023-05-25,
2023-05-26, 2023-05-27, 2023-05-28, 2023-05-29, 2023-05-30, 2023-05-31}
```

To adjust to the population size of the province:

```
In[]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "ON"][[1, 1]]]]
Out[]= {ON, 15 109 416, 1 918 812}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[]:= per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
       per100000A = 100 000 / popsize[[3]]
```

Out[]= $\frac{25\,000}{3\,297\,651}$

Out[]= $\frac{25\,000}{479\,703}$

```
In[]:= Clear[datetabON];
       datetabON[i_] = {0, 0};
       For[i = 1, i ≤ Length[counts], i++,
        date = Position[dates, counts[[i, 1]]][[1, 1]];
        If[
          ((counts[[i, 3]] == "70s") || (counts[[i, 3]] == "80s") || (counts[[i, 3]] == "90+")),
          datetabON[date] = datetabON[date] + {0, per100000A},
          datetabON[date] = datetabON[date] + {per100000Y, 0}
        ]
       ]
```

Processing ON Case Data [ENTER this or the previous folder - pre-processed data]

```
In[]:= startafter = "2021-11-30";
```

The two periods we’ll consider:

```
In[]:= mystartdate1 = "2021-12-01";
       myenddate1 = "2022-08-31";
       mystartdate2 = "2022-09-01";
       myenddate2 = "2023-05-31";

In[]:= curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

In[∘]:=
```
xaxis = {{curdate["2021-10-01"], "Oct"},
   {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
   {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
   {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
   {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
   {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
   {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
   {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
   {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
   {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
   {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

To adjust to the population size of the province:

In[∘]:=
```
popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "ON"][[1, 1]]]]
```

Out[∘]=
```
{ON, 15 109 416, 1 918 812}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

In[∘]:=
```
per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
per100000A = 100 000 / popsize[[3]]
```

Out[∘]= $\frac{25\,000}{3\,297\,651}$

Out[∘]= $\frac{25\,000}{479\,703}$

In[∘]:=
```
tabdates = {"2021-12-01", "2021-12-02", "2021-12-03",
   "2021-12-04", "2021-12-05", "2021-12-06", "2021-12-07",
   "2021-12-08", "2021-12-09", "2021-12-10", "2021-12-11", "2021-12-12",
   "2021-12-13", "2021-12-14", "2021-12-15", "2021-12-16", "2021-12-17",
   "2021-12-18", "2021-12-19", "2021-12-20", "2021-12-21", "2021-12-22",
   "2021-12-23", "2021-12-24", "2021-12-25", "2021-12-26", "2021-12-27",
   "2021-12-28", "2021-12-29", "2021-12-30", "2021-12-31", "2022-01-01",
   "2022-01-02", "2022-01-03", "2022-01-04", "2022-01-05", "2022-01-06",
   "2022-01-07", "2022-01-08", "2022-01-09", "2022-01-10", "2022-01-11",
   "2022-01-12", "2022-01-13", "2022-01-14", "2022-01-15", "2022-01-16",
   "2022-01-17", "2022-01-18", "2022-01-19", "2022-01-20", "2022-01-21",
   "2022-01-22", "2022-01-23", "2022-01-24", "2022-01-25", "2022-01-26",
   "2022-01-27", "2022-01-28", "2022-01-29", "2022-01-30", "2022-01-31",
   "2022-02-01", "2022-02-02", "2022-02-03", "2022-02-04", "2022-02-05",
   "2022-02-06", "2022-02-07", "2022-02-08", "2022-02-09", "2022-02-10",
   "2022-02-11", "2022-02-12", "2022-02-13", "2022-02-14", "2022-02-15",
   "2022-02-16", "2022-02-17", "2022-02-18", "2022-02-19", "2022-02-20",
   "2022-02-21", "2022-02-22", "2022-02-23", "2022-02-24", "2022-02-25",
```

```
"2022-02-26", "2022-02-27", "2022-02-28", "2022-03-01", "2022-03-02",
"2022-03-03", "2022-03-04", "2022-03-05", "2022-03-06", "2022-03-07",
"2022-03-08", "2022-03-09", "2022-03-10", "2022-03-11", "2022-03-12",
"2022-03-13", "2022-03-14", "2022-03-15", "2022-03-16", "2022-03-17",
"2022-03-18", "2022-03-19", "2022-03-20", "2022-03-21", "2022-03-22",
"2022-03-23", "2022-03-24", "2022-03-25", "2022-03-26", "2022-03-27",
"2022-03-28", "2022-03-29", "2022-03-30", "2022-03-31", "2022-04-01",
"2022-04-02", "2022-04-03", "2022-04-04", "2022-04-05", "2022-04-06",
"2022-04-07", "2022-04-08", "2022-04-09", "2022-04-10", "2022-04-11",
"2022-04-12", "2022-04-13", "2022-04-14", "2022-04-15", "2022-04-16",
"2022-04-17", "2022-04-18", "2022-04-19", "2022-04-20", "2022-04-21",
"2022-04-22", "2022-04-23", "2022-04-24", "2022-04-25", "2022-04-26",
"2022-04-27", "2022-04-28", "2022-04-29", "2022-04-30", "2022-05-01",
"2022-05-02", "2022-05-03", "2022-05-04", "2022-05-05", "2022-05-06",
"2022-05-07", "2022-05-08", "2022-05-09", "2022-05-10", "2022-05-11",
"2022-05-12", "2022-05-13", "2022-05-14", "2022-05-15", "2022-05-16",
"2022-05-17", "2022-05-18", "2022-05-19", "2022-05-20", "2022-05-21",
"2022-05-22", "2022-05-23", "2022-05-24", "2022-05-25", "2022-05-26",
"2022-05-27", "2022-05-28", "2022-05-29", "2022-05-30", "2022-05-31",
"2022-06-01", "2022-06-02", "2022-06-03", "2022-06-04", "2022-06-05",
"2022-06-06", "2022-06-07", "2022-06-08", "2022-06-09", "2022-06-10",
"2022-06-11", "2022-06-12", "2022-06-13", "2022-06-14", "2022-06-15",
"2022-06-16", "2022-06-17", "2022-06-18", "2022-06-19", "2022-06-20",
"2022-06-21", "2022-06-22", "2022-06-23", "2022-06-24", "2022-06-25",
"2022-06-26", "2022-06-27", "2022-06-28", "2022-06-29", "2022-06-30",
"2022-07-01", "2022-07-02", "2022-07-03", "2022-07-04", "2022-07-05",
"2022-07-06", "2022-07-07", "2022-07-08", "2022-07-09", "2022-07-10",
"2022-07-11", "2022-07-12", "2022-07-13", "2022-07-14", "2022-07-15",
"2022-07-16", "2022-07-17", "2022-07-18", "2022-07-19", "2022-07-20",
"2022-07-21", "2022-07-22", "2022-07-23", "2022-07-24", "2022-07-25",
"2022-07-26", "2022-07-27", "2022-07-28", "2022-07-29", "2022-07-30",
"2022-07-31", "2022-08-01", "2022-08-02", "2022-08-03", "2022-08-04",
"2022-08-05", "2022-08-06", "2022-08-07", "2022-08-08", "2022-08-09",
"2022-08-10", "2022-08-11", "2022-08-12", "2022-08-13", "2022-08-14",
"2022-08-15", "2022-08-16", "2022-08-17", "2022-08-18", "2022-08-19",
"2022-08-20", "2022-08-21", "2022-08-22", "2022-08-23", "2022-08-24",
"2022-08-25", "2022-08-26", "2022-08-27", "2022-08-28", "2022-08-29",
"2022-08-30", "2022-08-31", "2022-09-01", "2022-09-02", "2022-09-03",
"2022-09-04", "2022-09-05", "2022-09-06", "2022-09-07", "2022-09-08",
"2022-09-09", "2022-09-10", "2022-09-11", "2022-09-12", "2022-09-13",
"2022-09-14", "2022-09-15", "2022-09-16", "2022-09-17", "2022-09-18",
"2022-09-19", "2022-09-20", "2022-09-21", "2022-09-22", "2022-09-23",
"2022-09-24", "2022-09-25", "2022-09-26", "2022-09-27", "2022-09-28",
"2022-09-29", "2022-09-30", "2022-10-01", "2022-10-02", "2022-10-03",
```

```
"2022-10-04", "2022-10-05", "2022-10-06", "2022-10-07", "2022-10-08",
"2022-10-09", "2022-10-10", "2022-10-11", "2022-10-12", "2022-10-13",
"2022-10-14", "2022-10-15", "2022-10-16", "2022-10-17", "2022-10-18",
"2022-10-19", "2022-10-20", "2022-10-21", "2022-10-22", "2022-10-23",
"2022-10-24", "2022-10-25", "2022-10-26", "2022-10-27", "2022-10-28",
"2022-10-29", "2022-10-30", "2022-10-31", "2022-11-01", "2022-11-02",
"2022-11-03", "2022-11-04", "2022-11-05", "2022-11-06", "2022-11-07",
"2022-11-08", "2022-11-09", "2022-11-10", "2022-11-11", "2022-11-12",
"2022-11-13", "2022-11-14", "2022-11-15", "2022-11-16", "2022-11-17",
"2022-11-18", "2022-11-19", "2022-11-20", "2022-11-21", "2022-11-22",
"2022-11-23", "2022-11-24", "2022-11-25", "2022-11-26", "2022-11-27",
"2022-11-28", "2022-11-29", "2022-11-30", "2022-12-01", "2022-12-02",
"2022-12-03", "2022-12-04", "2022-12-05", "2022-12-06", "2022-12-07",
"2022-12-08", "2022-12-09", "2022-12-10", "2022-12-11", "2022-12-12",
"2022-12-13", "2022-12-14", "2022-12-15", "2022-12-16", "2022-12-17",
"2022-12-18", "2022-12-19", "2022-12-20", "2022-12-21", "2022-12-22",
"2022-12-23", "2022-12-24", "2022-12-25", "2022-12-26", "2022-12-27",
"2022-12-28", "2022-12-29", "2022-12-30", "2022-12-31", "2023-01-01",
"2023-01-02", "2023-01-03", "2023-01-04", "2023-01-05", "2023-01-06",
"2023-01-07", "2023-01-08", "2023-01-09", "2023-01-10", "2023-01-11",
"2023-01-12", "2023-01-13", "2023-01-14", "2023-01-15", "2023-01-16",
"2023-01-17", "2023-01-18", "2023-01-19", "2023-01-20", "2023-01-21",
"2023-01-22", "2023-01-23", "2023-01-24", "2023-01-25", "2023-01-26",
"2023-01-27", "2023-01-28", "2023-01-29", "2023-01-30", "2023-01-31",
"2023-02-01", "2023-02-02", "2023-02-03", "2023-02-04", "2023-02-05",
"2023-02-06", "2023-02-07", "2023-02-08", "2023-02-09", "2023-02-10",
"2023-02-11", "2023-02-12", "2023-02-13", "2023-02-14", "2023-02-15",
"2023-02-16", "2023-02-17", "2023-02-18", "2023-02-19", "2023-02-20",
"2023-02-21", "2023-02-22", "2023-02-23", "2023-02-24", "2023-02-25",
"2023-02-26", "2023-02-27", "2023-02-28", "2023-03-01", "2023-03-02",
"2023-03-03", "2023-03-04", "2023-03-05", "2023-03-06", "2023-03-07",
"2023-03-08", "2023-03-09", "2023-03-10", "2023-03-11", "2023-03-12",
"2023-03-13", "2023-03-14", "2023-03-15", "2023-03-16", "2023-03-17",
"2023-03-18", "2023-03-19", "2023-03-20", "2023-03-21", "2023-03-22",
"2023-03-23", "2023-03-24", "2023-03-25", "2023-03-26", "2023-03-27",
"2023-03-28", "2023-03-29", "2023-03-30", "2023-03-31", "2023-04-01",
"2023-04-02", "2023-04-03", "2023-04-04", "2023-04-05", "2023-04-06",
"2023-04-07", "2023-04-08", "2023-04-09", "2023-04-10", "2023-04-11",
"2023-04-12", "2023-04-13", "2023-04-14", "2023-04-15", "2023-04-16",
"2023-04-17", "2023-04-18", "2023-04-19", "2023-04-20", "2023-04-21",
"2023-04-22", "2023-04-23", "2023-04-24", "2023-04-25", "2023-04-26",
"2023-04-27", "2023-04-28", "2023-04-29", "2023-04-30", "2023-05-01",
"2023-05-02", "2023-05-03", "2023-05-04", "2023-05-05", "2023-05-06",
"2023-05-07", "2023-05-08", "2023-05-09", "2023-05-10", "2023-05-11",
```

```
"2023-05-12", "2023-05-13", "2023-05-14", "2023-05-15", "2023-05-16",
"2023-05-17", "2023-05-18", "2023-05-19", "2023-05-20", "2023-05-21",
"2023-05-22", "2023-05-23", "2023-05-24", "2023-05-25", "2023-05-26",
"2023-05-27", "2023-05-28", "2023-05-29", "2023-05-30", "2023-05-31"};
```

tab = Table[datetabON[i], {i, 1, Length[dates]}] // N

```
In[◦]:= tab = {{8.41508091668888`, 3.8565529087789736`},
  {8.142159373444915`, 4.58617102665607`}, {9.036735542966797`,
   3.3353971102953284`}, {7.171771664133045`, 2.762125731963319`},
  {8.2634578371089`, 2.970588051356777`}, {11.765950975406433`,
   4.117130808020796`}, {12.008547902734401`, 4.221361967717526`},
  {12.599877913096321`, 4.27347754756589`}, {11.925155208965412`,
   4.429824287110983`}, {14.351124482245089`, 4.4819398669593475`},
  {14.078202939001125`, 3.074819211053506`}, {18.54350263263153`,
   4.58617102665607`}, {29.801516291445033`, 5.1073268251397215`},
  {30.498982457512938`, 5.993291682561918`}, {33.084155964351595`,
   6.879256539984115`}, {34.50183175842441`, 6.722909800439021`},
  {42.69705920972231`, 9.172342053312153`}, {46.04034811446087`,
   8.703301834676873`}, {54.174926333926784`, 8.755417414525237`},
  {86.14465266336553`, 15.061402576177343`}, {88.46448578093921`,
   20.48142288040725`}, {87.81250653874531`, 18.24045294692758`},
  {93.45288509912055`, 21.99277469600982`}, {85.49267342117162`,
   21.888543536313094`}, {55.19838212109165`, 18.03199062753412`},
  {80.64831602859127`, 28.611453336752117`}, {130.7370003678376`,
   42.31785083687198`}, {159.90928087902572`, 53.21000702518016`},
  {183.1910047485316`, 59.25541428759045`}, {182.4328893506317`,
   63.476776255307975`}, {109.16103614360647`, 48.10268020004044`},
  {53.826193250892835`, 34.44839827976894`}, {71.80110933509944`,
   42.839006635355624`}, {92.80848701090564`, 58.31733385031988`},
  {92.82364931886364`, 62.27811791879559`}, {87.35763730000536`,
   59.30752986743881`}, {76.10720479517087`, 60.55830378379956`},
  {70.34552777113163`, 58.0046403712297`}, {54.40236095329676`,
   42.89112221520399`}, {51.8550932163531`, 43.25593127414254`},
  {70.47440738877462`, 63.89370089409489`}, {59.99725258979801`,
   56.44117297577876`}, {52.514653612526004`, 58.63002732941007`},
  {47.260913905079704`, 53.26212260502853`}, {45.88872503488089`,
   48.67595157837245`}, {34.90363291931135`, 36.793599372945344`},
  {31.5527628605938`, 32.93704646416637`}, {30.931108234315882`,
   34.81320733870749`}, {40.29383339837963`, 44.194011711413104`},
  {36.74585333620811`, 47.5815244015568`}, {31.029663236042868`,
   38.20072002885119`}, {30.19573629835298`, 36.42879031400679`},
  {24.84344158917969`, 26.474714562969172`}, {21.742749611769106`,
   21.315272157981084`}, {30.19573629835298`, 44.08978055171637`},
  {26.397578154874484`, 31.060885589625247`}, {23.89579734180482`,
```

28.1945286979652`}, {21.0225399837642`, 25.849327604788797`},
{19.938434964767346`, 24.077397889944404`}, {16.883229911230753`,
14.644477937390427`}, {16.640632983902783`, 15.634673954509353`},
{22.508446163648003`, 26.5789457226659`}, {20.81026767235223`,
26.526830142817534`}, {18.88465456168648 5`, 18.761608745411223`},
{16.61030836798679`, 17.09391019026356`}, {17.436654151697677`,
17.66718156859557`}, {13.843187165652157`, 10.05830691073435`},
{14.214663710623107`, 8.963879733918695`}, {18.68754455823251`,
16.729101131325006`}, {16.579983752070792`, 14.22755329860351`},
{14.207082556644108`, 11.204849667398369`}, {13.289762925185231`,
11.465427566640193`}, {12.721176376760306`, 10.735809448763089`},
{10.280044795522631`, 7.5567590780128535`}, {11.386893276456483`,
7.713105817557947`}, {14.282894096434099`, 11.621774306185285`},
{13.092652921731256`, 9.224457633160519`}, {12.872799456340285`,
13.602166340423137`}, {11.326244044624492`, 9.641382271947434`},
{11.000254423527535`, 10.318884809976172`}, {10.151165177879648`,
7.035603279529209`}, {10.43924902908161`, 6.045407262410283`},
{12.000966748755403`, 8.182146036193227`}, {14.366286790203088`,
11.726005465882015`}, {12.85005599440329`, 8.130030456344864`},
{10.780400958136564`, 7.504643498164489`}, {10.446830183060609`,
7.660990237709583`}, {9.370306318042752`, 6.8271409601357504`},
{10.393762105207616`, 5.1073268251397215`}, {13.555103314450195`,
8.338492775738322`}, {12.599877913096321`, 9.901960171189257`},
{11.303500582687494`, 8.963879733918695`}, {10.264882487564632`,
5.889060522865189`}, {9.673552477202712`, 5.993291682561918`},
{9.279332470294765`, 5.4721358840782734`}, {10.340694027354623`,
6.253869581803741`}, {13.456548312723209`, 8.286377195889957`},
{12.98651676602527`, 7.400412338467761`}, {11.470285970225472`,
7.765221397406312`}, {10.378599797249619`, 7.504643498164489`},
{10.499898260913602`, 7.244065598922667`}, {9.097384774798789`,
5.4721358840782734`}, {10.113259407984653`, 5.1073268251397215`},
{14.116108708896121`, 10.52734712936963`}, {12.304212907915362`,
7.9215681369514055`}, {11.94789867090241`, 9.901960171189257`},
{10.795563266094563`, 7.660990237709583`}, {11.803856745301427`,
8.286377195889957`}, {10.939605191695543`, 6.93137211983248`},
{13.047165997857263`, 7.035603279529209`}, {17.451816459655674`,
12.768317062849304`}, {16.38287374861682`, 12.403508003910753`},
{14.169176786749114`, 11.152734087550005`}, {13.828024857694158`,
10.944271768156547`}, {15.31393103757796`, 11.830236625578744`},
{12.645364836970316`, 12.403508003910753`}, {15.124402188102986`,
10.63157828906636`}, {21.356110758840156`, 17.615065988747205`},
{21.34094845088216`, 15.582558374660989`}, {19.695838037439376`,
18.13622178723085`}, {18.346392629177558`, 19.75180476253015`},
{20.362979587591287`, 19.75180476253015`}, {17.914266852374613`,

15.426211635115894`}, {19.415335340216416`, 19.07430222450141`},
{26.480970848643473`, 26.37048340327244`}, {24.7600488954107`,
22.20123701540328`}, {22.516027317627003`, 25.797212024940432`},
{21.227231141197173`, 24.75490042797314`}, {21.265136911092167`,
28.298759857661928`}, {17.80054954268963`, 21.471618897526177`},
{19.726162653355374`, 21.62796563707127`}, {25.859316222365557`,
36.324559154310066`}, {23.380278871232886`, 31.060885589625247`},
{21.037702291722198`, 30.435498631444872`}, {18.952884947497477`,
36.116096834916604`}, {14.972779108523007`, 27.51702615993646`},
{15.306349883598962`, 23.973166730247673`}, {16.102371051393856`,
25.953558764485525`}, {20.393304203507284`, 35.80340335582642`},
{21.136257293449187`, 37.78379539006427`}, {18.209931857555574`,
35.17801639764604`}, {15.829449508149892`, 34.44839827976894`},
{15.260862959724967`, 32.78069972462128`}, {12.501322911369336`,
28.038181958420108`}, {12.69085176084431`, 23.764704410854215`},
{16.95146029704174`, 39.5036095250603`}, {15.185051419934977`,
30.74819211053506`}, {12.721176376760306`, 29.34107145462922`},
{11.94789867090241`, 26.891639201756085`}, {11.242851350855503`,
29.862227253112863`}, {9.059479004903794`, 16.98967903056683`},
{10.901699421800549`, 23.712588831005853`}, {13.054747151836262`,
35.85551893567478`}, {12.281469445978365`, 28.715684496448844`},
{10.530222876829598`, 27.308563840543002`}, {9.787269786887697`,
21.0025786788909`}, {8.991248619092802`, 23.556242091460756`},
{7.596316286956989`, 14.852940256783885`}, {8.452986686583875`,
14.33178445830024`}, {11.311081736666495`, 25.171825066760057`},
{9.810013248824694`, 20.68988519980071`}, {8.900274771344815`,
21.315272157981084`}, {7.755520520515967`, 18.657377585714496`},
{7.907143600095947`, 18.24045294692758`}, {5.92846241157721`,
13.028894962091128`}, {6.489467806023136`, 12.35139242406239`},
{7.945049369990942`, 21.36738773782945`}, {6.678996655498111`,
16.207945332841362`}, {6.30752011052716`, 12.82043264269767`},
{5.2537397074463`, 12.976779382242762`}, {4.57901700331539`,
14.279668878451876`}, {4.290933152113428`, 9.276573213008882`},
{4.192378150386442`, 8.494839515283415`}, {4.525948925462397`,
8.651186254828508`}, {5.685865484249243`, 10.579462709217996`},
{5.625216252417251`, 14.123322138906781`}, {5.026305088076331`,
8.963879733918695`}, {4.290933152113428`, 10.318884809976172`},
{3.8133204514364922`, 6.462331901197199`}, {3.9270377611214773`,
8.651186254828508`}, {5.132441243782316`, 10.944271768156547`},
{5.011142780118333`, 10.423115969672901`}, {4.776127006769364`,
10.371000389824538`}, {4.480462001588403`, 7.348296758619396`},
{4.3440012299664215`, 7.869452557103041`}, {3.7147654497095055`,
5.889060522865189`}, {3.3963569825915476`, 6.410216321348835`},
{5.193090475614309`, 9.432919952553975`}, {4.442556231693408`,

```
 7.504643498164489`}, {4.1544723804914465`, 8.130030456344864`},
{3.760252373583499`, 7.191950019074302`}, {3.851226221331487`,
 7.191950019074302`}, {2.547267736943661`, 5.1073268251397215`},
{3.15376005526358`, 5.836944943016825`}, {4.5865981572943895`,
 8.963879733918695`}, {4.1544723804914465`, 7.97368371679977`},
{4.048336224785461`, 8.390608355586686`}, {3.6616973718565125`,
 7.452527918316124`}, {3.7299277576675034`, 8.338492775738322`},
{3.434262752486543`, 7.139834439225938`}, {3.7829958355204965`,
 6.410216321348835`}, {5.526661250690264`, 11.726005465882015`},
{4.927750086349344`, 11.882352205427107`}, {5.033886242055329`,
 12.35139242406239`}, {4.950493548286341`, 12.299276844214024`},
{4.510786617504399`, 12.71620148300094`}, {4.162053534470446`,
 9.276573213008882`}, {4.753383544832367`, 10.05830691073435`},
{7.240002049944036`, 15.530442794812624`}, {6.876106658952084`,
 11.204849667398369`}, {6.034598567283196`, 11.986583365123836`},
{6.61076626968712`, 13.602166340423137`}, {5.723771254144238`,
 12.664085903152575`}, {6.117991261052185`, 10.475231549521267`},
{6.845782043036088`, 14.019090979210054`}, {9.044316696945796`,
 17.146025770111923`}, {8.650096690037849`, 17.302372509657015`},
{8.748651691764834`, 18.81372432525959`}, {7.6266409028729845`,
 18.81372432525959`}, {7.80858859836896`, 18.917955484956316`},
{6.565279345813126`, 15.37409605526753`}, {7.156609356175047`,
 21.732196796768`}, {11.356568660540487`, 26.318367823424076`},
{10.36343748929162`, 23.868935570550946`}, {10.211814409711641`,
 25.223940646608423`}, {8.47573014852087`, 29.393187034477585`},
{8.862369001449819`, 27.569141739784826`}, {7.975373985906938`,
 20.064498241620335`}, {8.18764629731891`, 24.233744629489497`},
{11.356568660540487`, 36.58513705355188`}, {10.234557871648637`,
 29.810111673264498`}, {10.28762594950163`, 32.25954392613763`},
{9.0139920810298`, 25.276056226456788`}, {9.180777468567777`,
 24.85913158766987`}, {7.13386589423805`, 20.377191720710524`},
{7.042892046490062`, 19.491226863288325`}, {9.98437979034167`,
 30.174920732203052`}, {8.885112463386816`, 25.64086528539534`},
{8.430243224646878`, 26.422598983120807`}, {8.164902835381913`,
 24.70278484812478`}, {7.702452442662974`, 25.588749705546974`},
{6.0725043371781915`, 17.510834829050474`}, {6.648672039582115`,
 19.386995703591598`}, {8.210389759255907`, 21.471618897526177`},
{9.817594402803692`, 29.966458412809594`}, {8.650096690037849`,
 28.09029753826847`}, {7.816169752347959`, 25.06759390706333`},
{7.619059748893985`, 24.02528231009604`}, {6.133153569010183`,
 16.468523232083186`}, {6.375750496338151`, 18.448915266321038`},
{9.650809015265715`, 29.288955874780854`}, {7.899562446116948`,
 23.295664192218936`}, {7.929887062032944`, 25.64086528539534`},
{7.36130051360802`, 21.575850057222908`}, {7.391625129524016`,
```

```
20.898347519194168`}, {5.564567020585259`, 16.676985551476644`},
{6.216546262779172`, 16.103714173144635`}, {9.08980362081979`,
 26.787408042059358`}, {7.596316286956989`, 25.276056226456788`},
{7.5584105170619935`, 22.566046074341834`}, {6.648672039582115`,
 23.868935570550946`}, {6.800295119162095`, 19.022186644653047`},
{5.473593172837271`, 17.250256929808653`}, {5.92846241157721`,
 15.843136273902811`}, {8.885112463386816`, 28.246644277813562`},
{7.080797816385057`, 25.588749705546974`}, {6.724483579372104`,
 21.732196796768`}, {5.8829754877032165`, 21.106809838587626`},
{6.337844726443156`, 23.81681999070258`}, {4.692734313000375`,
 15.686789534357716`}, {5.079373165929324`, 14.905055836632249`},
{7.907143600095947`, 25.223940646608423`}, {6.8154574271200925`,
 21.83642795646473`}, {5.716190100165239`, 21.99277469600982`},
{5.473593172837271`, 17.719297148443932`}, {5.193090475614309`,
 19.491226863288325`}, {4.139310072533449`, 14.22755329860351`},
{4.28335199813443`, 13.550050760574772`}, {4.632085081168383`,
 16.312176492538093`}, {6.6941589634561085`, 21.575850057222908`},
{5.549404712627261`, 21.680081216919636`}, {5.314388939278293`,
 16.41640765223482`}, {5.428106248963277`, 18.084106207382483`},
{4.632085081168383`, 15.47832721496426`}, {5.0187239340973315`,
 15.634673954509353`}, {6.876106658952084`, 22.930855133280385`},
{5.77683933199723`, 22.67027723403856`}, {5.594891636501255`,
 17.56295040889884`}, {5.685865484249243`, 22.87873955343202`},
{5.314388939278293`, 18.084106207382483`}, {4.609341619231386`,
 14.019090979210054`}, {5.155184705719313`, 17.45871924920211`},
{7.535667055124996`, 21.62796563707127`}, {6.345425880422155`,
 22.826623973583654`}, {6.034598567283196`, 22.357583754948376`},
{5.920881257598212`, 18.917955484956316`}, {5.2537397074463`,
 18.81372432525959`}, {4.404650461798414`, 15.94736743359954`},
{5.026305088076331`, 18.60526200586613`}, {7.042892046490062`,
 29.13260913523576`}, {6.489467806023136`, 27.204332680846274`},
{5.9512058735142075`, 25.171825066760057`}, {5.170347013677311`,
 26.21413666372735`}, {4.867100854517352`, 21.211040998284354`},
{4.162053534470446`, 17.510834829050474`}, {4.768545852790365`,
 20.585654040103982`}, {7.217258588007039`, 33.770895741740205`},
{6.428818574191144`, 29.080493555387395`}, {6.277195494611164`,
 31.52992580826053`}, {6.148315876968181`, 33.82301132158857`},
{6.057342029220194`, 32.20742834628927`}, {4.783708160748363`,
 23.45201093176403`}, {4.6396662351473825`, 20.898347519194168`},
{6.133153569010183`, 31.113001169473613`}, {8.051185525696928`,
 41.223423660056326`}, {7.467436669314005`, 33.35397110295329`},
{6.421237420212145`, 35.75128777597806`}, {5.8829754877032165`,
 32.78069972462128`}, {4.662409697084379`, 22.409699334796738`},
{5.0869543199083225`, 24.598553688428048`}, {7.459855515335007`,
```

```
 38.30495118854791`}, {6.292357802569162`, 35.85551893567478`},
{6.064923183199192`, 32.1553127664409`}, {6.1255724150311845`,
 31.58204138810889`}, {6.269614340632165`, 31.946850447047446`},
{4.791289314727362`, 23.295664192218936`}, {5.0187239340973315`,
 27.82971963902665`}, {7.179352818112044`, 38.096488869154456`},
{6.860944350994086`, 30.85242327023179`}, {5.875394333724218`,
 34.81320733870749`}, {6.337844726443156`, 32.10319718659254`},
{5.321970093257291`, 26.109905504030618`}, {3.90429429918448`,
 18.709493165562858`}, {4.253027382218433`, 20.898347519194168`},
{6.16347818492618`, 28.142413118116835`}, {5.860232025766219`,
 27.204332680846274`}, {5.147603551740315`, 28.976262395690668`},
{4.594179311273388`, 25.06759390706333`}, {4.328838922008424`,
 22.30546817510001`}, {3.30538313484356`, 17.35448808950538`},
{3.434262752486543`, 17.041794610415195`}, {4.723058928916371`,
 21.732196796768`}, {3.957362377037473`, 18.709493165562858`},
{3.6616973718565125`, 16.05159859329627`}, {3.214409287095572`,
 17.92775946783739`}, {3.275058518927564`, 15.217749315722436`},
{2.8429327421246215`, 13.654281920271501`}, {2.547267736943661`,
 13.133126121787857`}, {3.82848275939449`, 18.396799686472672`},
{3.6920219877725082`, 16.833332291021737`}, {3.30538313484356`,
 18.29256852677594`}, {2.7671212023346317`, 17.250256929808653`},
{2.6534038926496466`, 12.24716126436566`}, {2.365320041447685`,
 10.371000389824538`}, {2.1757911919727104`, 11.934467785275473`},
{3.896713145205481`, 15.686789534357716`}, {3.494911984318535`,
 17.979875047685756`}, {3.3357077507595556`, 14.123322138906781`},
{3.07794851547359`, 13.393704021029679`}, {3.1234354393475843`,
 14.019090979210054`}, {2.623079276733651`, 11.413311986791827`},
{2.7747023563136306`, 10.16253807043108`}, {4.025592762848464`,
 15.634673954509353`}, {3.6541162178775135`, 14.748709097087156`},
{3.358451212696553`, 14.383900038148605`}, {3.237152749032569`,
 13.914859819513325`}, {3.5555612161505263`, 15.113518156025707`},
{2.7140531244816386`, 10.52734712936963`}, {2.5775923528596567`,
 10.52734712936963`}, {4.723058928916371`, 17.35448808950538`},
{3.995268146932468`, 14.488131197845334`}, {4.002849300911467`,
 16.155829752992997`}, {3.5252366002345306`, 14.644477937390427`},
{3.889131991226482`, 14.905055836632249`}, {2.949068897830607`,
 10.371000389824538`}, {3.282639672906563`, 12.038698944972202`},
{5.670703176291245`, 18.81372432525959`}, {5.003561626139334`,
 19.960267081923607`}, {4.419812769756412`, 20.012382661771973`},
{4.040755070806462`, 16.62486997162828`}, {4.480462001588403`,
 17.406603669353746`}, {3.366032366675552`, 14.905055836632249`},
{3.282639672906563`, 15.895251853751175`}, {5.352294709173287`,
 22.67027723403856`}, {5.246158553467302`, 23.087201872825478`},
{5.049048550013327`, 20.585654040103982`}, {4.381906999861417`,
```

```
 16.62486997162828`}, {3.30538313484356`, 15.999483013447904`},
{2.767121202334631 7`, 12.142930104668931`}, {3.1234354393475843`,
 12.924663802394399`}, {4.442556231693408`, 19.126417804349774`},
{4.973237010223338`, 20.79411635949744`}, {5.875394333724218`,
 25.53663412569861`}, {5.3826193250892835`, 23.660473251157487`},
{5.344713555194288`, 20.742000779649075`}, {3.9270377611214773`,
 17.87564388798903`}, {3.8057392974574933`, 18.344684106624307`},
{4.859519700538353`, 22.097005855706552`}, {6.262033186653166`,
 23.712588831005853`}, {4.8974254704333475`, 22.87873955343202`},
{4.57901700331539`, 19.022186644653047`}, {4.556273541378394`,
 23.45201093176403`}, {3.176503517200577`, 17.719297148443932`},
{3.290220826885562`, 16.364292072386455`}, {4.624503927189385`,
 20.168729401317066`}, {4.199959304365441`, 17.198141349960288`},
{3.767833527562498`, 18.13622178723085`}, {3.237152749032569`,
 17.041794610415195`}, {3.30538313484356`, 16.155829752992997`},
{2.7292154324396365`, 13.18524170163622`}, {2.8126081262086253`,
 12.403508003910753`}, {4.2227027663024375`, 17.66718156859557`},
{3.6010481400245204`, 18.97007106480468`}, {3.7375089116465023`,
 16.885447870870102`}, {3.282639672906563`, 13.706397500119866`},
{3.3963569825915476`, 14.436015617996969`}, {2.8732573580406173`,
 9.901960171189257`}, {2.5927546608176546`, 10.266769230127808`},
{4.3440012299664215`, 15.217749315722436`}, {3.843645067352488`,
 13.081010541939492`}, {3.5328177542135295`, 12.82043264269767`},
{3.4494250604445407`, 13.966975399361688`}, {3.5328177542135295`,
 11.830236625578744`}, {2.7595400483556327`, 10.006191330885986`},
{2.8808385120196163`, 9.954075751037621`}, {4.108985456617453`,
 14.123322138906781`}, {3.9346189151004762`, 11.569658726336922`},
{3.7223466036885045`, 13.497935180726408`}, {3.0476238995575944`,
 12.664085903152575`}, {3.350870058717554`, 12.82043264269767`},
{2.47903735113267`, 10.371000389824538`}, {2.7595400483556327`,
 10.423115969672901`}, {3.5555612161505263`, 12.24716126436566`},
{3.4645873684025386`, 13.602166340423137`}, {3.426681598507544`,
 13.237357281484586`}, {2.9263254358936104`, 12.195045684517295`},
{3.070367361494591`, 12.768317062849304`}, {2.380482349405683`,
 10.110422490582716`}, {2.471456197153671`, 10.475231549521267`},
{3.6541162178775135`, 13.550050760574772`}, {3.131016593326583`,
 13.75851307996823`}, {3.07794851547359`, 12.299276844214024`},
{2.6534038926496466`, 12.35139242406239`}, {2.532105428985663`,
 11.465427566640193`}, {2.228859269825703`, 9.48503553240234`},
{2.0544927283087264`, 10.318884809976172`}, {2.7898646642716285`,
 11.830236625578744`}, {3.358451212696553`, 13.810628659816595`},
{3.0552050535365933`, 13.341588441181313`}, {2.532105428985663`,
 11.048502927853276`}, {2.297089655636694`, 11.67388988603365`},
{2.0923984982037216`, 7.504643498164489`}, {1.9104508027077456`,
```

```
 10.371000389824538`}, {3.30538313484356`, 14.748709097087156`},
{2.888419665998615`, 13.497935180726408`}, {2.8050269722296264`,
 11.882352205427107`}, {2.463875043174672`, 12.142930104668931`},
{2.623079276733651`, 10.63157828906636`}, {1.8346392629177557`,
 8.182146036193227`}, {2.0544927283087264`, 11.10061850770164`},
{3.168922363221578`, 10.892156188308183`}, {2.949068897830607`,
 11.934467785275473`}, {2.607916968775653`, 10.683693868914725`},
{2.7064719705026397`, 10.318884809976172`}, {2.395644657363681`,
 7.869452557103041`}, {1.8498015708757536`, 8.494839515283415`},
{1.4328381020308092`, 6.514447481045564`}, {3.0627862075155923`,
 13.654281920271501`}, {2.7216342784606375`, 10.318884809976172`},
{2.7974458182506274`, 8.599070674980144`}, {2.2591838857416993`,
 11.204849667398369`}, {2.0469115743297275`, 9.745613431644163`},
{1.683016183337776`, 8.234261616041593`}, {1.9635188805607386`,
 8.599070674980144`}, {3.237152749032569`, 10.735809448763089`},
{2.32741427155269`, 11.256965247246734`}, {2.342576579510688`,
 9.797729011492528`}, {2.0999796521827205`, 8.911764154070331`},
{1.8801261867917496`, 8.025799296648135`}, {1.3873511781568153`,
 7.869452557103041`}, {1.4555815639678062`, 6.8271409601357504`},
{2.0544927283087264`, 9.12022647346379`}, {2.0014246504557334`,
 10.944271768156547`}, {1.3494454082618204`, 8.963879733918695`},
{1.622366951505784`, 9.12022647346379`}, {1.4934873338628012`,
 8.703301834676873`}, {0.9324819394168758`, 6.879256539984115`},
{1.1599165587868456`, 7.348296758619396`}, {1.675435029358777`,
 8.182146036193227`}, {1.3039584843878265`, 8.599070674980144`},
{1.2205657906188374`, 8.182146036193227`}, {1.3115396383668254`,
 7.348296758619396`}, {0.82634578371089`, 5.680598203471732`},
{0.788440013815895`, 7.296181178771032`}, {0.7505342439209001`,
 6.305985161652106`}, {1.3039584843878265`, 8.234261616041593`},
{1.5313931037577961`, 8.911764154070331`}, {1.2887961764298284`,
 10.52734712936963`}, {1.2660527144928315`, 9.797729011492528`},
{1.0310369411438627`, 8.911764154070331`}, {0.803602321773893`,
 6.983487699680844`}, {0.5155184705719313`, 6.305985161652106`},
{1.3267019463248233`, 8.599070674980144`}, {0.9173196314588778`,
 6.462331901197199`}, {0.9021573235008798`, 5.680598203471732`},
{0.8794138615638829`, 6.097522842258647`}, {0.7202096280049041`,
 5.836944943016825`}, {0.6292357802569162`, 4.8467489258978995`},
{0.6519792421939132`, 5.576367043775003`}, {0.9097384774798789`,
 6.410216321348835`}, {0.8869950155428818`, 6.35810074150047`},
{0.758115397899899`, 6.149638422107012`}, {0.82634578371089`,
 5.524251463926638`}, {0.6595603961729122`, 6.253869581803741`},
{0.5306807785299293`, 5.211557984836451`}, {0.5534242404669263`,
 5.263673564684815`}, {0.8339269376898889`, 7.452527918316124`},
{0.85667039962688 6`, 5.941176102713554`}, {0.796021167794894`,
```

```
6.149638422107012`}, {0.811183475752892`, 6.514447481045564`},
{0.5837488563829223`, 7.035603279529209`}, {0.45486923873993945`,
 4.534055446807712`}, {0.5685865484249243`, 5.1073268251397215`},
{0.6974661660679071`, 5.836944943016825`}, {0.7277907819839031`,
 7.191950019074302`}, {0.5458430864879273`, 5.732713783320096`},
{0.6292357802569162`, 5.211557984836451`}, {0.5306807785299293`,
 5.367904724381544`}, {0.4700315466979374`, 5.159442404988086`},
{0.3790576989499495`, 5.263673564684815`}, {0.796021167794894`,
 6.8271409601357504`}, {0.7050473200469061`, 5.836944943016825`},
{0.5306807785299293`, 6.097522842258647`}, {0.6368169342359152`,
 4.8467489258978995`}, {0.5837488563829223`, 5.263673564684815`},
{0.2880838512019616`, 4.794633346049535`}, {0.3866388529289485`,
 3.908668488627338`}, {0.5003561626139333`, 4.794633346049535`},
{0.758115397899899`, 6.722909800439021`}, {0.6671415501519111`,
 7.244065598922667`}, {0.6443980882149142`, 5.1073268251397215`},
{0.5079373165929324`, 4.586171026656077`}, {0.34115192905495456`,
 3.3353971102953284`}, {0.28050269722296267`, 4.429824287110983`},
{0.7429530899419011`, 5.4721358840782734`}, {0.6368169342359152`,
 5.4721358840782734`}, {0.6216546262779172`, 5.367904724381544`}};
```

Enter the following to avoid upload:

```
In[]:= dates = tabdates;
```

```
In[]:= For[i = 1, i <= Length[dates], i++, datetabON[i] = tab[[i]]]
```

Overall Analysis by 70+

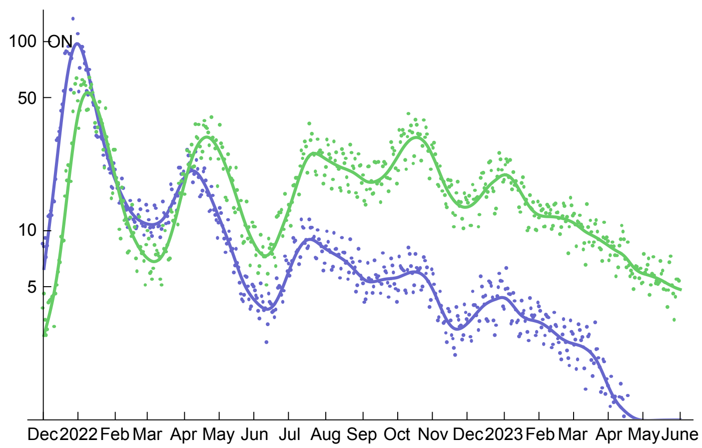
Cubic Spline fit (https://mathematica.stackexchange.com/questions/33206/implementation-of-smoothing-splines-function)

```
In[]:= CubicSplSmooth[data_, lambda_] := Module[{M, Knots, X, Dsq, a}, M = Length@data;
   Knots = Flatten@{1, 1, 1, Range@M, M, M, M};
   X = Table[Evaluate@N@BSplineBasis[{3, Knots}, n, t], {t, 1, M}, {n, 0, M + 1}];
   Dsq = Differences[X, 2];
   a = LinearSolve[Transpose[X].X + lambda * Transpose[Dsq].Dsq,
     Transpose[X].data, Method → "Multifrontal"];
   Return[
    X.
     a]];
```

Cubic spline fit to all dates (Y is younger, A is 70+), using lambda = 3 to provide a good fit but not overfit to the data:

```
In[∘]:= modYON = CubicSplSmooth[Table[{i, Log[Max[1, datetabON[i][[1]]]]},
            {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
        modAON = CubicSplSmooth[Table[{i, Log[Max[1, datetabON[i][[2]]]]},
            {i, 1, Length[dates]}], 10^lambda /. lambda → 3];

In[∘]:= plotON = Show[
          ListPlot[Table[{i, Log[datetabON[i][[1]]]}, {i, 1, Length[dates]}],
           PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.4, 0.8]}],
          ListLinePlot[modYON, PlotStyle → RGBColor[0.4, 0.4, 0.8], PlotRange → All],
          ListLinePlot[modAON, PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
          ListPlot[Table[{i, Log[datetabON[i][[2]]]}, {i, 1, Length[dates]}],
           PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
          Graphics[Text["ON", {5, 4.6}, {-1, 0}]],
          PlotRange → {Automatic, {0, 5}}, AxesOrigin → {1, 0},
          Ticks → {xaxis, {{Log[5], "5"}, {Log[10], "10"}, {Log[50], "50"}, {Log[100], "100"},
             {Log[500], "500"}, {Log[1000], "1000"}, {Log[5000], "5000"}}}
         ]
```

Out[∘]=

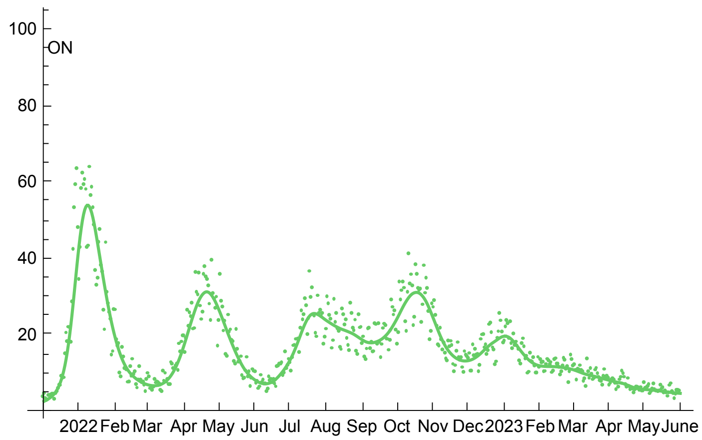


The above illustrates the major drop off in reported cases among those <70:

70+ on a non-log plot

In[∘]:=
```
Show[
 ListLinePlot[Table[{modAON[[i, 1]], Exp[modAON[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabON[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["ON", {5, 95}, {-1, 0}]],
 PlotRange → {Automatic, {0, 100}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

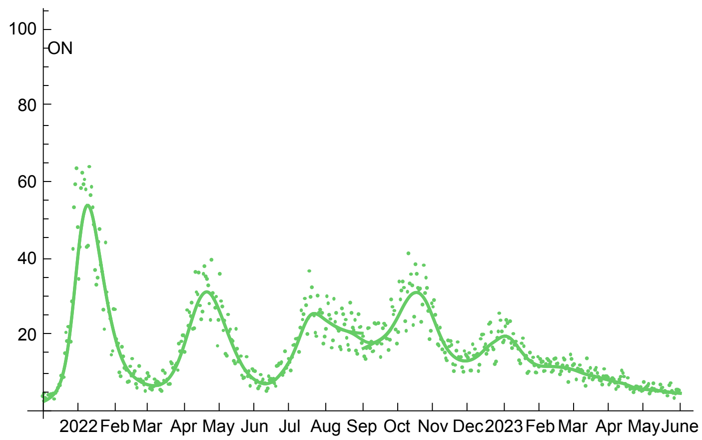
Out[∘]=



In[∘]:=
```
modAON1 = CubicSplSmooth[Table[{i, Log[Max[1, datetabON[i][[2]]]]},
    {i, 1, Position[dates, myenddate1][[1, 1]]}], 10^lambda /. lambda → 3];
modAON2 = CubicSplSmooth[Table[{i, Log[Max[1, datetabON[i][[2]]]]},
    {i, Position[dates, mystartdate2][[1, 1]], Length[dates]}],
   10^lambda /. lambda → 3];
```

In[•]:=
```
Show[
 ListLinePlot[Table[{modAON[[i, 1]], Exp[modAON[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAON1[[i, 1]], Exp[modAON1[[i, 2]]]},
   {i, 1, Length[modAON1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAON2[[i, 1]], Exp[modAON2[[i, 2]]]},
   {i, 1, Length[modAON2]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabON[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["ON", {5, 95}, {-1, 0}]],
 PlotRange → {Automatic, {0, 100}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

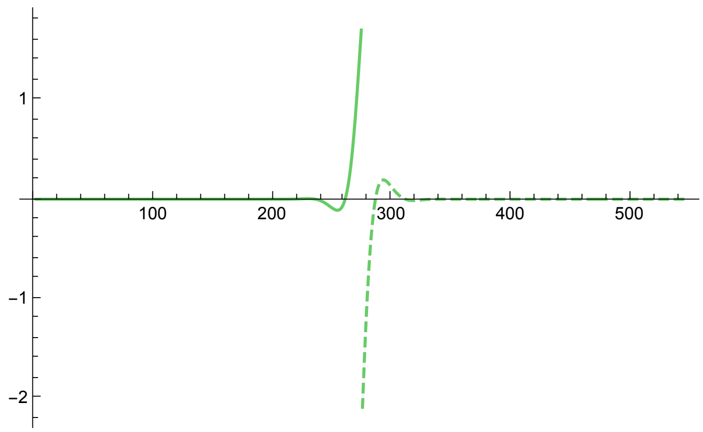
Out[•]=



The fit is very similar, with a slight discrepancy in numbers at the joint, so we just use the overall spline fit for the analysis:

In[∘]:=
```
Show[
 ListLinePlot[Table[{modAON1[[i, 1]], Exp[modAON1[[i, 2]]] - Exp[modAON[[i, 2]]]},
   {i, 1, Length[modAON1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAON2[[i, 1]], Exp[modAON2[[i, 2]]] - Exp[modAON[[
       i + Position[dates, myenddate1][[1, 1]], 2]]]}, {i, 1, Length[modAON2]}],
  PlotStyle → {Dashed, RGBColor[0.4, 0.8, 0.4]}, PlotRange → All],
 PlotRange → All
]
```

Out[∘]=



Merging with Omicron subvariant frequencies

In[∘]:=
```
graphheight = 63;
```

In[∘]:=
```
colourBA1 = RGBColor[0.5, 0.1, 0.1];
colourBA11 = RGBColor[0.75, 0.1, 0.1];
colourBA2 = RGBColor[1, 0.1, 0.1];
colourBA5 = RGBColor[1, 0.5, 0.1];
colourBA45 = colourBA5;
colourBA4 = colourBA5;
colourBQ = RGBColor[1, 0.75, 0.5];
colourXBB15 = RGBColor[0.2, 0.5, 0.8];
colourXBBO = RGBColor[0.5, 0.7, 0.8];

{colourBA1, colourBA11, colourBA2, colourBA45, colourBQ, colourXBB15, colourXBBO}
```

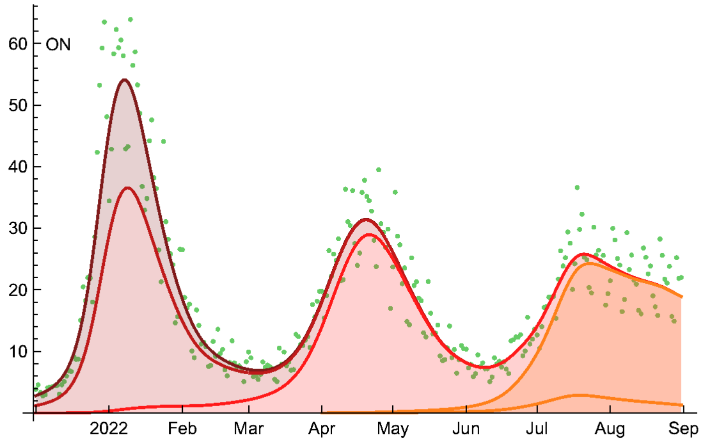
Out[∘]= {■, ■, ■, ■, ■, ■, ■}

From Omicron_VariantsGISAID.nb (selection on BA.1, BA.1.1 and BA.2 calculated from 16 May analysis; selection on BA.4&BA.5 from 24 June analysis where the initial frequency is for “2022-04-1” at t=1):

In[∘]:=
```
ONML = {pBA11 → 0.4179983246932652`, pBA2 → 0.0006274609768064037`,
   sBA11 → 0.026697737809944225`, sBA2 → 0.09026330895189581`,
   pBA4 → 0.0001836350901127891`, pBA5 → 0.0002664111940827573`,
   sBA4 → 0.08164738439327635`, sBA5 → 0.09622847239814021`};
```

In[◦]:=
```
endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modAON[[i, 1]],
    1 * Exp[modAON[[i, 2]]] /. t -> modAON[[i, 1]] /. ONML}, {i, 1, endtime}];
tabBA11 = Table[{modAON[[i, 1]],
    (pBA2 Exp[sBA2 t] + pBA11 Exp[sBA11 t])/((1 - pBA11 - pBA2) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t]) * Exp[modAON[[i, 2]]] /.
      t -> modAON[[i, 1]] /. ONML}, {i, 1, endtime}];
tabBA2 = Table[{modAON[[i, 1]],
    (pBA2 Exp[sBA2 t])/((1 - pBA11 - pBA2) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t]) * Exp[modAON[[i, 2]]] /.
      t -> modAON[[i, 1]] /. ONML}, {i, 1, endtime}];
tabBA4 = Table[{modAON[[i, 1]], (pBA4 Exp[sBA4 t])/((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
       Exp[modAON[[i, 2]]] /. t -> modAON[[i, 1]] - curdate["2022-03-31"] /.
     ONML}, {i, curdate["2022-04-1"], endtime}];
tabBA5 = Table[{modAON[[i, 1]], (pBA5 Exp[sBA5 t] + pBA4 Exp[sBA4 t])/((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
       Exp[modAON[[i, 2]]] /. t -> modAON[[i, 1]] - curdate["2022-03-31"] /.
     ONML}, {i, curdate["2022-04-1"], endtime}];

Show[
 ListLinePlot[Table[{modAON[[i, 1]], Exp[modAON[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabON[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA4, tabBA5}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
     Filling → {1 → {2}, 2 → {3}, 3 → Axis, 4 → Axis, 5 → {4}}],
 Graphics[Text["ON", {5, 0.95 * graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[•]=



From duotang/VirusSeq (see code above): mutantNames = list("BA.1.1*", "BA.2*", "BA.4/5*", "BA.1*")

In[•]:=
```
ONML1
```

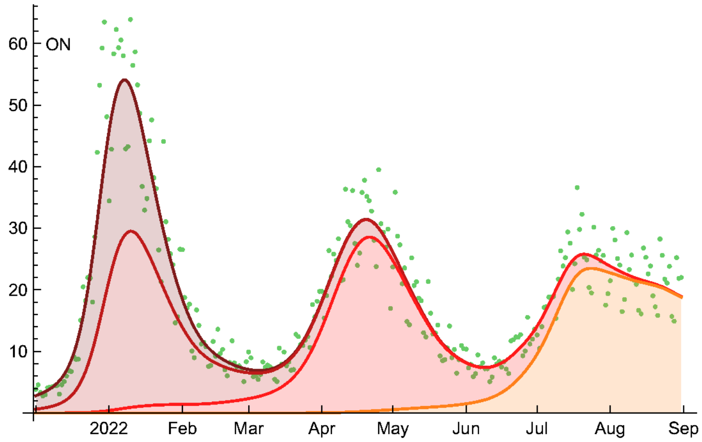
Out[•]=
```
{0.118989, 0.872082, 0.00602749, 0.0370314, 0.0943605, 0.168613}
```

In[•]:=
```
dataname = ONML1;
refdate = "2022-04-15"; (* t=0 date used in duotang code above *)
ONML = {pBA11 → dataname[[1]], pBA2 → dataname[[2]], pBA45 → dataname[[3]],
   sBA11 → dataname[[4]], sBA2 → dataname[[5]], sBA45 → dataname[[6]]}
```

Out[•]=
```
{pBA11 → 0.118989, pBA2 → 0.872082, pBA45 → 0.00602749,
 sBA11 → 0.0370314, sBA2 → 0.0943605, sBA45 → 0.168613}
```

```
In[∘]:= endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modAON[[i, 1]],
    1 * Exp[modAON[[i, 2]]] /. t -> modAON[[i, 1]] /. ONML}, {i, 1, endtime}];
tabBA11 = Table[{modAON[[i, 1]],
    (pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAON[[i, 2]]] /.
     t -> modAON[[i, 1]] - curdate[refdate] /. ONML}, {i, 1, endtime}];
tabBA2 = Table[{modAON[[i, 1]],
    (pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAON[[i, 2]]] /.
     t -> modAON[[i, 1]] - curdate[refdate] /. ONML}, {i, 1, endtime}];
tabBA45 = Table[{modAON[[i, 1]],
    (pBA45 Exp[sBA45 t]) /
    ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAON[[i, 2]]] /.
     t -> modAON[[i, 1]] - curdate[refdate] /. ONML}, {i, 1, endtime}];

Show[
 ListLinePlot[Table[{modAON[[i, 1]], Exp[modAON[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabON[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
     Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis}],
 Graphics[Text["ON", {5, 0.95 graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[◦]=



From duotang/VirusSeq (see code above): mutantNames = list(“XBB.1.5*”, “XBB.Other*”, “BQ*”, “BA.5*”)

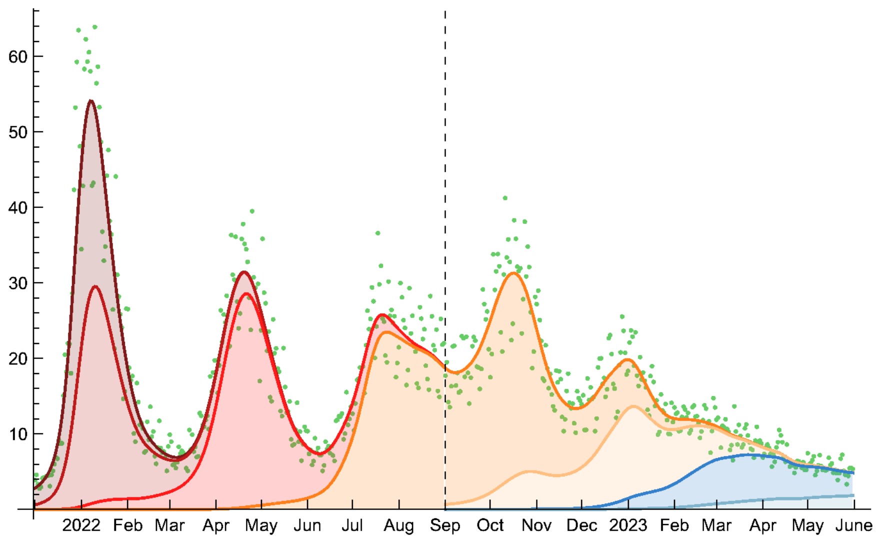
In[◦]:=
```
ONML2
```

Out[◦]=
```
{0.129499, 0.00762521, 0.634858, 0.0743729, 0.0914175, 0.0320328}
```

In[◦]:=
```
dataname = ONML2;
refdate = "2023-01-14"; (* t=0 date used in duotang code above *)
ONML = {pXBB15 → dataname[[1]], pXBBO → dataname[[2]], pBQ → dataname[[3]],
  sXBB15 → dataname[[4]], sXBBO → dataname[[5]], sBQ → dataname[[6]]}
```

Out[◦]=
```
{pXBB15 → 0.129499, pXBBO → 0.00762521, pBQ → 0.634858,
 sXBB15 → 0.0743729, sXBBO → 0.0914175, sBQ → 0.0320328}
```

```
In[•]:= starttime = Position[dates, "2022-09-01"][[1, 1]];
endtime = Length[dates];

tabBA5 = Table[{modAON[[i, 1]], 1 * Exp[modAON[[i, 2]]] /. t -> modAON[[i, 1]] /. ONML},
    {i, starttime, endtime}];
tabBQ = Table[{modAON[[i, 1]],
      (pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
      ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
        Exp[modAON[[i, 2]]] /.
      t -> modAON[[i, 1]] - curdate[refdate] /. ONML}, {i, starttime, endtime}];
tabXBB15 = Table[{modAON[[i, 1]],
      (pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
      ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
        Exp[modAON[[i, 2]]] /.
      t -> modAON[[i, 1]] - curdate[refdate] /. ONML}, {i, starttime, endtime}];
tabXBBO = Table[{modAON[[i, 1]],
      (pXBBO Exp[sXBBO t]) /
      ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
        Exp[modAON[[i, 2]]] /.
      t -> modAON[[i, 1]] - curdate[refdate] /. ONML}, {i, starttime, endtime}];

Show[
 ListLinePlot[Table[{modAON[[i, 1]], Exp[modAON[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabON[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45, tabBA5, tabBQ, tabXBB15, tabXBBO},
  PlotStyle → {colourBA1, colourBA11, colourBA2, colourBA45,
    colourBA45, colourBQ, colourXBB15, colourXBBO}, PlotRange → All,
      Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis,
    5 → {6}, 6 → {7}, 7 → {8}, 8 → Axis}],
 (*Graphics[Text["ON",{5,95},{-1,0}]],*)
 ListPlot[{{starttime, 0}, {starttime, 100}},
  Joined → True, PlotStyle → {Black, Thickness[0.002], Dashed}],
 PlotRange → {{1, curdate["2023-06-01"]}, {0, graphheight}},
 Ticks → {xaxis, Automatic},
 AspectRatio → 0.6,
 ImageSize → 450
]
```

Out[ ]=


AB - 70+ case counts and genomic frequencies (Dec 2021-May 2023)

https://www.alberta.ca/stats/covid-19-alberta-statistics.htm#data-export

Processing AB Case Data [ENTER this or the next folder - uploading & processing data]

In[ ]:=
```
SetDirectory["/Users/otto/Dropbox/COVID19/BC_Data/CorrectingCaseCounts"];
```

In[ ]:=
```
SetDirectory["/Users/otto/Downloads"];
```

In[ ]:=
```
startafter = "2021-11-30";
```

The two periods we’ll consider:

In[ ]:=
```
mystartdate1 = "2021-12-01";
myenddate1 = "2022-08-31";
mystartdate2 = "2022-09-01";
myenddate2 = "2023-05-31";
```

In[ ]:=
```
curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

```
xaxis = {{curdate["2021-10-01"], "Oct"},
   {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
   {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
   {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
   {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
   {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
   {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
   {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
   {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
   {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
   {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

Uploading the data (see below to avoid this step):

```
In[∘]:= temp0 = Import["covid-19-alberta-statistics-data9June2023.csv"];
temp0[[1]]
temp = Drop[temp0, 1];
counts = Transpose[{temp[[All, 2]], temp[[All, 3]], temp[[All, 5]]}];
Clear[temp0, temp]

Out[∘]= {, Date reported, Alberta Health Services Zone,
 Gender, Age group, Death status, Case type}

In[∘]:= counts = Sort[counts];

In[∘]:= Last[Position[counts, startafter]][[1]]

Out[∘]= 335 915

In[∘]:= counts = counts[[% + 1 ;; Length[counts]]];

In[∘]:= dates = DeleteDuplicates[counts[[All, 1]]]

Out[∘]= {2021-12-01, 2021-12-02, 2021-12-03, 2021-12-04, 2021-12-05, 2021-12-06, 2021-12-07,
 2021-12-08, 2021-12-09, 2021-12-10, 2021-12-11, 2021-12-12, 2021-12-13, 2021-12-14,
 2021-12-15, 2021-12-16, 2021-12-17, 2021-12-18, 2021-12-19, 2021-12-20, 2021-12-21,
 2021-12-22, 2021-12-23, 2021-12-24, 2021-12-25, 2021-12-26, 2021-12-27, 2021-12-28,
 2021-12-29, 2021-12-30, 2021-12-31, 2022-01-01, 2022-01-02, 2022-01-03,
 2022-01-04, 2022-01-05, 2022-01-06, 2022-01-07, 2022-01-08, 2022-01-09,
 2022-01-10, 2022-01-11, 2022-01-12, 2022-01-13, 2022-01-14, 2022-01-15,
 2022-01-16, 2022-01-17, 2022-01-18, 2022-01-19, 2022-01-20, 2022-01-21,
 2022-01-22, 2022-01-23, 2022-01-24, 2022-01-25, 2022-01-26, 2022-01-27,
 2022-01-28, 2022-01-29, 2022-01-30, 2022-01-31, 2022-02-01, 2022-02-02,
 2022-02-03, 2022-02-04, 2022-02-05, 2022-02-06, 2022-02-07, 2022-02-08,
 2022-02-09, 2022-02-10, 2022-02-11, 2022-02-12, 2022-02-13, 2022-02-14,
 2022-02-15, 2022-02-16, 2022-02-17, 2022-02-18, 2022-02-19, 2022-02-20,
 2022-02-21, 2022-02-22, 2022-02-23, 2022-02-24, 2022-02-25, 2022-02-26,
 2022-02-27, 2022-02-28, 2022-03-01, 2022-03-02, 2022-03-03, 2022-03-04,
 2022-03-05, 2022-03-06, 2022-03-07, 2022-03-08, 2022-03-09, 2022-03-10,
```

```
2022-03-11, 2022-03-12, 2022-03-13, 2022-03-14, 2022-03-15, 2022-03-16,
2022-03-17, 2022-03-18, 2022-03-19, 2022-03-20, 2022-03-21, 2022-03-22,
2022-03-23, 2022-03-24, 2022-03-25, 2022-03-26, 2022-03-27, 2022-03-28,
2022-03-29, 2022-03-30, 2022-03-31, 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04,
2022-04-05, 2022-04-06, 2022-04-07, 2022-04-08, 2022-04-09, 2022-04-10,
2022-04-11, 2022-04-12, 2022-04-13, 2022-04-14, 2022-04-15, 2022-04-16,
2022-04-17, 2022-04-18, 2022-04-19, 2022-04-20, 2022-04-21, 2022-04-22,
2022-04-23, 2022-04-24, 2022-04-25, 2022-04-26, 2022-04-27, 2022-04-28,
2022-04-29, 2022-04-30, 2022-05-01, 2022-05-02, 2022-05-03, 2022-05-04,
2022-05-05, 2022-05-06, 2022-05-07, 2022-05-08, 2022-05-09, 2022-05-10,
2022-05-11, 2022-05-12, 2022-05-13, 2022-05-14, 2022-05-15, 2022-05-16,
2022-05-17, 2022-05-18, 2022-05-19, 2022-05-20, 2022-05-21, 2022-05-22,
2022-05-23, 2022-05-24, 2022-05-25, 2022-05-26, 2022-05-27, 2022-05-28,
2022-05-29, 2022-05-30, 2022-05-31, 2022-06-01, 2022-06-02, 2022-06-03,
2022-06-04, 2022-06-05, 2022-06-06, 2022-06-07, 2022-06-08, 2022-06-09,
2022-06-10, 2022-06-11, 2022-06-12, 2022-06-13, 2022-06-14, 2022-06-15,
2022-06-16, 2022-06-17, 2022-06-18, 2022-06-19, 2022-06-20, 2022-06-21,
2022-06-22, 2022-06-23, 2022-06-24, 2022-06-25, 2022-06-26, 2022-06-27,
2022-06-28, 2022-06-29, 2022-06-30, 2022-07-01, 2022-07-02, 2022-07-03, 2022-07-04,
2022-07-05, 2022-07-06, 2022-07-07, 2022-07-08, 2022-07-09, 2022-07-10,
2022-07-11, 2022-07-12, 2022-07-13, 2022-07-14, 2022-07-15, 2022-07-16,
2022-07-17, 2022-07-18, 2022-07-19, 2022-07-20, 2022-07-21, 2022-07-22,
2022-07-23, 2022-07-24, 2022-07-25, 2022-07-26, 2022-07-27, 2022-07-28,
2022-07-29, 2022-07-30, 2022-07-31, 2022-08-01, 2022-08-02, 2022-08-03,
2022-08-04, 2022-08-05, 2022-08-06, 2022-08-07, 2022-08-08, 2022-08-09,
2022-08-10, 2022-08-11, 2022-08-12, 2022-08-13, 2022-08-14, 2022-08-15,
2022-08-16, 2022-08-17, 2022-08-18, 2022-08-19, 2022-08-20, 2022-08-21,
2022-08-22, 2022-08-23, 2022-08-24, 2022-08-25, 2022-08-26, 2022-08-27,
2022-08-28, 2022-08-29, 2022-08-30, 2022-08-31, 2022-09-01, 2022-09-02,
2022-09-03, 2022-09-04, 2022-09-05, 2022-09-06, 2022-09-07, 2022-09-08,
2022-09-09, 2022-09-10, 2022-09-11, 2022-09-12, 2022-09-13, 2022-09-14,
2022-09-15, 2022-09-16, 2022-09-17, 2022-09-18, 2022-09-19, 2022-09-20,
2022-09-21, 2022-09-22, 2022-09-23, 2022-09-24, 2022-09-25, 2022-09-26,
2022-09-27, 2022-09-28, 2022-09-29, 2022-09-30, 2022-10-01, 2022-10-02,
2022-10-03, 2022-10-04, 2022-10-05, 2022-10-06, 2022-10-07, 2022-10-08,
2022-10-09, 2022-10-10, 2022-10-11, 2022-10-12, 2022-10-13, 2022-10-14,
2022-10-15, 2022-10-16, 2022-10-17, 2022-10-18, 2022-10-19, 2022-10-20,
2022-10-21, 2022-10-22, 2022-10-23, 2022-10-24, 2022-10-25, 2022-10-26,
2022-10-27, 2022-10-28, 2022-10-29, 2022-10-30, 2022-10-31, 2022-11-01,
2022-11-02, 2022-11-03, 2022-11-04, 2022-11-05, 2022-11-06, 2022-11-07,
2022-11-08, 2022-11-09, 2022-11-10, 2022-11-11, 2022-11-12, 2022-11-13,
2022-11-14, 2022-11-15, 2022-11-16, 2022-11-17, 2022-11-18, 2022-11-19,
2022-11-20, 2022-11-21, 2022-11-22, 2022-11-23, 2022-11-24, 2022-11-25,
2022-11-26, 2022-11-27, 2022-11-28, 2022-11-29, 2022-11-30, 2022-12-01,
```

```
 2022-12-02, 2022-12-03, 2022-12-04, 2022-12-05, 2022-12-06, 2022-12-07,
 2022-12-08, 2022-12-09, 2022-12-10, 2022-12-11, 2022-12-12, 2022-12-13,
 2022-12-14, 2022-12-15, 2022-12-16, 2022-12-17, 2022-12-18, 2022-12-19,
 2022-12-20, 2022-12-21, 2022-12-22, 2022-12-23, 2022-12-24, 2022-12-25,
 2022-12-26, 2022-12-27, 2022-12-28, 2022-12-29, 2022-12-30, 2022-12-31,
 2023-01-01, 2023-01-02, 2023-01-03, 2023-01-04, 2023-01-05, 2023-01-06,
 2023-01-07, 2023-01-08, 2023-01-09, 2023-01-10, 2023-01-11, 2023-01-12,
 2023-01-13, 2023-01-14, 2023-01-15, 2023-01-16, 2023-01-17, 2023-01-18,
 2023-01-19, 2023-01-20, 2023-01-21, 2023-01-22, 2023-01-23, 2023-01-24,
 2023-01-25, 2023-01-26, 2023-01-27, 2023-01-28, 2023-01-29, 2023-01-30,
 2023-01-31, 2023-02-01, 2023-02-02, 2023-02-03, 2023-02-04, 2023-02-05,
 2023-02-06, 2023-02-07, 2023-02-08, 2023-02-09, 2023-02-10, 2023-02-11,
 2023-02-12, 2023-02-13, 2023-02-14, 2023-02-15, 2023-02-16, 2023-02-17,
 2023-02-18, 2023-02-19, 2023-02-20, 2023-02-21, 2023-02-22, 2023-02-23,
 2023-02-24, 2023-02-25, 2023-02-26, 2023-02-27, 2023-02-28, 2023-03-01,
 2023-03-02, 2023-03-03, 2023-03-04, 2023-03-05, 2023-03-06, 2023-03-07,
 2023-03-08, 2023-03-09, 2023-03-10, 2023-03-11, 2023-03-12, 2023-03-13,
 2023-03-14, 2023-03-15, 2023-03-16, 2023-03-17, 2023-03-18, 2023-03-19,
 2023-03-20, 2023-03-21, 2023-03-22, 2023-03-23, 2023-03-24, 2023-03-25,
 2023-03-26, 2023-03-27, 2023-03-28, 2023-03-29, 2023-03-30, 2023-03-31,
 2023-04-01, 2023-04-02, 2023-04-03, 2023-04-04, 2023-04-05, 2023-04-06,
 2023-04-07, 2023-04-08, 2023-04-09, 2023-04-10, 2023-04-11, 2023-04-12,
 2023-04-13, 2023-04-14, 2023-04-15, 2023-04-16, 2023-04-17, 2023-04-18,
 2023-04-19, 2023-04-20, 2023-04-21, 2023-04-22, 2023-04-23, 2023-04-24,
 2023-04-25, 2023-04-26, 2023-04-27, 2023-04-28, 2023-04-29, 2023-04-30,
 2023-05-01, 2023-05-02, 2023-05-03, 2023-05-04, 2023-05-05, 2023-05-06,
 2023-05-07, 2023-05-08, 2023-05-09, 2023-05-10, 2023-05-11, 2023-05-12,
 2023-05-13, 2023-05-14, 2023-05-15, 2023-05-16, 2023-05-17, 2023-05-18,
 2023-05-19, 2023-05-20, 2023-05-21, 2023-05-22, 2023-05-23, 2023-05-24,
 2023-05-25, 2023-05-26, 2023-05-27, 2023-05-28, 2023-05-29, 2023-05-30,
 2023-05-31, 2023-06-01, 2023-06-02, 2023-06-03, 2023-06-04, 2023-06-05}
```

Trim off data from Alberta after end date (this data stream keeps going for a few more weeks):

```
In[∘]:= Last[Position[counts, myenddate2]][[1]]
Out[∘]= 297 334
```

```
In[∘]:= counts = counts[[1 ;; %]];
```

```
In[∘]:= dates = DeleteDuplicates[counts[[All, 1]]]
Out[∘]= {2021-12-01, 2021-12-02, 2021-12-03, 2021-12-04, 2021-12-05, 2021-12-06, 2021-12-07,
 2021-12-08, 2021-12-09, 2021-12-10, 2021-12-11, 2021-12-12, 2021-12-13, 2021-12-14,
 2021-12-15, 2021-12-16, 2021-12-17, 2021-12-18, 2021-12-19, 2021-12-20, 2021-12-21,
 2021-12-22, 2021-12-23, 2021-12-24, 2021-12-25, 2021-12-26, 2021-12-27, 2021-12-28,
 2021-12-29, 2021-12-30, 2021-12-31, 2022-01-01, 2022-01-02, 2022-01-03,
```

2022-01-04, 2022-01-05, 2022-01-06, 2022-01-07, 2022-01-08, 2022-01-09,
2022-01-10, 2022-01-11, 2022-01-12, 2022-01-13, 2022-01-14, 2022-01-15,
2022-01-16, 2022-01-17, 2022-01-18, 2022-01-19, 2022-01-20, 2022-01-21,
2022-01-22, 2022-01-23, 2022-01-24, 2022-01-25, 2022-01-26, 2022-01-27,
2022-01-28, 2022-01-29, 2022-01-30, 2022-01-31, 2022-02-01, 2022-02-02,
2022-02-03, 2022-02-04, 2022-02-05, 2022-02-06, 2022-02-07, 2022-02-08,
2022-02-09, 2022-02-10, 2022-02-11, 2022-02-12, 2022-02-13, 2022-02-14,
2022-02-15, 2022-02-16, 2022-02-17, 2022-02-18, 2022-02-19, 2022-02-20,
2022-02-21, 2022-02-22, 2022-02-23, 2022-02-24, 2022-02-25, 2022-02-26,
2022-02-27, 2022-02-28, 2022-03-01, 2022-03-02, 2022-03-03, 2022-03-04,
2022-03-05, 2022-03-06, 2022-03-07, 2022-03-08, 2022-03-09, 2022-03-10,
2022-03-11, 2022-03-12, 2022-03-13, 2022-03-14, 2022-03-15, 2022-03-16,
2022-03-17, 2022-03-18, 2022-03-19, 2022-03-20, 2022-03-21, 2022-03-22,
2022-03-23, 2022-03-24, 2022-03-25, 2022-03-26, 2022-03-27, 2022-03-28,
2022-03-29, 2022-03-30, 2022-03-31, 2022-04-01, 2022-04-02, 2022-04-03, 2022-04-04,
2022-04-05, 2022-04-06, 2022-04-07, 2022-04-08, 2022-04-09, 2022-04-10,
2022-04-11, 2022-04-12, 2022-04-13, 2022-04-14, 2022-04-15, 2022-04-16,
2022-04-17, 2022-04-18, 2022-04-19, 2022-04-20, 2022-04-21, 2022-04-22,
2022-04-23, 2022-04-24, 2022-04-25, 2022-04-26, 2022-04-27, 2022-04-28,
2022-04-29, 2022-04-30, 2022-05-01, 2022-05-02, 2022-05-03, 2022-05-04,
2022-05-05, 2022-05-06, 2022-05-07, 2022-05-08, 2022-05-09, 2022-05-10,
2022-05-11, 2022-05-12, 2022-05-13, 2022-05-14, 2022-05-15, 2022-05-16,
2022-05-17, 2022-05-18, 2022-05-19, 2022-05-20, 2022-05-21, 2022-05-22,
2022-05-23, 2022-05-24, 2022-05-25, 2022-05-26, 2022-05-27, 2022-05-28,
2022-05-29, 2022-05-30, 2022-05-31, 2022-06-01, 2022-06-02, 2022-06-03,
2022-06-04, 2022-06-05, 2022-06-06, 2022-06-07, 2022-06-08, 2022-06-09,
2022-06-10, 2022-06-11, 2022-06-12, 2022-06-13, 2022-06-14, 2022-06-15,
2022-06-16, 2022-06-17, 2022-06-18, 2022-06-19, 2022-06-20, 2022-06-21,
2022-06-22, 2022-06-23, 2022-06-24, 2022-06-25, 2022-06-26, 2022-06-27,
2022-06-28, 2022-06-29, 2022-06-30, 2022-07-01, 2022-07-02, 2022-07-03, 2022-07-04,
2022-07-05, 2022-07-06, 2022-07-07, 2022-07-08, 2022-07-09, 2022-07-10,
2022-07-11, 2022-07-12, 2022-07-13, 2022-07-14, 2022-07-15, 2022-07-16,
2022-07-17, 2022-07-18, 2022-07-19, 2022-07-20, 2022-07-21, 2022-07-22,
2022-07-23, 2022-07-24, 2022-07-25, 2022-07-26, 2022-07-27, 2022-07-28,
2022-07-29, 2022-07-30, 2022-07-31, 2022-08-01, 2022-08-02, 2022-08-03,
2022-08-04, 2022-08-05, 2022-08-06, 2022-08-07, 2022-08-08, 2022-08-09,
2022-08-10, 2022-08-11, 2022-08-12, 2022-08-13, 2022-08-14, 2022-08-15,
2022-08-16, 2022-08-17, 2022-08-18, 2022-08-19, 2022-08-20, 2022-08-21,
2022-08-22, 2022-08-23, 2022-08-24, 2022-08-25, 2022-08-26, 2022-08-27,
2022-08-28, 2022-08-29, 2022-08-30, 2022-08-31, 2022-09-01, 2022-09-02,
2022-09-03, 2022-09-04, 2022-09-05, 2022-09-06, 2022-09-07, 2022-09-08,
2022-09-09, 2022-09-10, 2022-09-11, 2022-09-12, 2022-09-13, 2022-09-14,
2022-09-15, 2022-09-16, 2022-09-17, 2022-09-18, 2022-09-19, 2022-09-20,
2022-09-21, 2022-09-22, 2022-09-23, 2022-09-24, 2022-09-25, 2022-09-26, 2022-09-27,

```
2022-09-28, 2022-09-29, 2022-09-30, 2022-10-01, 2022-10-02, 2022-10-03,
2022-10-04, 2022-10-05, 2022-10-06, 2022-10-07, 2022-10-08, 2022-10-09,
2022-10-10, 2022-10-11, 2022-10-12, 2022-10-13, 2022-10-14, 2022-10-15,
2022-10-16, 2022-10-17, 2022-10-18, 2022-10-19, 2022-10-20, 2022-10-21,
2022-10-22, 2022-10-23, 2022-10-24, 2022-10-25, 2022-10-26, 2022-10-27,
2022-10-28, 2022-10-29, 2022-10-30, 2022-10-31, 2022-11-01, 2022-11-02,
2022-11-03, 2022-11-04, 2022-11-05, 2022-11-06, 2022-11-07, 2022-11-08,
2022-11-09, 2022-11-10, 2022-11-11, 2022-11-12, 2022-11-13, 2022-11-14,
2022-11-15, 2022-11-16, 2022-11-17, 2022-11-18, 2022-11-19, 2022-11-20,
2022-11-21, 2022-11-22, 2022-11-23, 2022-11-24, 2022-11-25, 2022-11-26,
2022-11-27, 2022-11-28, 2022-11-29, 2022-11-30, 2022-12-01, 2022-12-02,
2022-12-03, 2022-12-04, 2022-12-05, 2022-12-06, 2022-12-07, 2022-12-08,
2022-12-09, 2022-12-10, 2022-12-11, 2022-12-12, 2022-12-13, 2022-12-14,
2022-12-15, 2022-12-16, 2022-12-17, 2022-12-18, 2022-12-19, 2022-12-20,
2022-12-21, 2022-12-22, 2022-12-23, 2022-12-24, 2022-12-25, 2022-12-26,
2022-12-27, 2022-12-28, 2022-12-29, 2022-12-30, 2022-12-31, 2023-01-01,
2023-01-02, 2023-01-03, 2023-01-04, 2023-01-05, 2023-01-06, 2023-01-07,
2023-01-08, 2023-01-09, 2023-01-10, 2023-01-11, 2023-01-12, 2023-01-13,
2023-01-14, 2023-01-15, 2023-01-16, 2023-01-17, 2023-01-18, 2023-01-19,
2023-01-20, 2023-01-21, 2023-01-22, 2023-01-23, 2023-01-24, 2023-01-25,
2023-01-26, 2023-01-27, 2023-01-28, 2023-01-29, 2023-01-30, 2023-01-31,
2023-02-01, 2023-02-02, 2023-02-03, 2023-02-04, 2023-02-05, 2023-02-06,
2023-02-07, 2023-02-08, 2023-02-09, 2023-02-10, 2023-02-11, 2023-02-12,
2023-02-13, 2023-02-14, 2023-02-15, 2023-02-16, 2023-02-17, 2023-02-18,
2023-02-19, 2023-02-20, 2023-02-21, 2023-02-22, 2023-02-23, 2023-02-24,
2023-02-25, 2023-02-26, 2023-02-27, 2023-02-28, 2023-03-01, 2023-03-02,
2023-03-03, 2023-03-04, 2023-03-05, 2023-03-06, 2023-03-07, 2023-03-08,
2023-03-09, 2023-03-10, 2023-03-11, 2023-03-12, 2023-03-13, 2023-03-14,
2023-03-15, 2023-03-16, 2023-03-17, 2023-03-18, 2023-03-19, 2023-03-20,
2023-03-21, 2023-03-22, 2023-03-23, 2023-03-24, 2023-03-25, 2023-03-26,
2023-03-27, 2023-03-28, 2023-03-29, 2023-03-30, 2023-03-31, 2023-04-01,
2023-04-02, 2023-04-03, 2023-04-04, 2023-04-05, 2023-04-06, 2023-04-07,
2023-04-08, 2023-04-09, 2023-04-10, 2023-04-11, 2023-04-12, 2023-04-13,
2023-04-14, 2023-04-15, 2023-04-16, 2023-04-17, 2023-04-18, 2023-04-19,
2023-04-20, 2023-04-21, 2023-04-22, 2023-04-23, 2023-04-24, 2023-04-25,
2023-04-26, 2023-04-27, 2023-04-28, 2023-04-29, 2023-04-30, 2023-05-01,
2023-05-02, 2023-05-03, 2023-05-04, 2023-05-05, 2023-05-06, 2023-05-07,
2023-05-08, 2023-05-09, 2023-05-10, 2023-05-11, 2023-05-12, 2023-05-13,
2023-05-14, 2023-05-15, 2023-05-16, 2023-05-17, 2023-05-18, 2023-05-19,
2023-05-20, 2023-05-21, 2023-05-22, 2023-05-23, 2023-05-24, 2023-05-25,
2023-05-26, 2023-05-27, 2023-05-28, 2023-05-29, 2023-05-30, 2023-05-31}
```

To adjust to the population size of the province:

```
In[◦]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "AB"][[1, 1]]]]
Out[◦]= {AB, 4 543 111, 439 109}
```

Age list (UNKNOWN treated as <70):

```
In[◦]:= DeleteDuplicates[counts[[All, 3]]]
Out[◦]= {10-19 years, 1-4 years, 20-29 years, 30-39 years, 40-49 years, 50-59 years,
  5-9 years, 60-69 years, 70-79 years, 80+ years, Under 1 year, Unknown}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[◦]:= per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
        per100000A = 100 000 / popsize[[3]]
```

Out[◦]= $\frac{50\,000}{2\,052\,001}$

Out[◦]= $\frac{100\,000}{439\,109}$

```
In[◦]:= Clear[datetabAB];
datetabAB[i_] = {0, 0};
For[i = 1, i ≤ Length[counts], i++,
 date = Position[dates, counts[[i, 1]]][[1, 1]];
 If[((counts[[i, 3]] == "70-79 years") || (counts[[i, 3]] == "80+ years")),
  datetabAB[date] = datetabAB[date] + {0, per100000A},
  datetabAB[date] = datetabAB[date] + {per100000Y, 0}
 ]
]
```

Processing AB Case Data [ENTER this or the previous folder - pre-processed data]

```
In[◦]:= startafter = "2021-11-30";
```

The two periods we’ll consider:

```
In[◦]:= mystartdate1 = "2021-12-01";
myenddate1 = "2022-08-31";
mystartdate2 = "2022-09-01";
myenddate2 = "2023-05-31";
```

```
In[◦]:= curdate[x_] := InputForm[DateObject[x] - DateObject[startafter]][[1, 1]];
```

```
In[∘]:= xaxis = {{curdate["2021-10-01"], "Oct"},
    {curdate["2021-11-01"], "Nov"}, {curdate["2021-12-01"], "Dec"},
    {curdate["2022-01-01"], "2022"}, {curdate["2022-02-01"], "Feb"},
    {curdate["2022-03-01"], "Mar"}, {curdate["2022-04-01"], "Apr"},
    {curdate["2022-05-01"], "May"}, {curdate["2022-06-01"], "Jun"},
    {curdate["2022-07-01"], "Jul"}, {curdate["2022-08-01"], "Aug"},
    {curdate["2022-09-01"], "Sep"}, {curdate["2022-10-01"], "Oct"},
    {curdate["2022-11-01"], "Nov"}, {curdate["2022-12-01"], "Dec"},
    {curdate["2023-01-01"], "2023"}, {curdate["2023-02-01"], "Feb"},
    {curdate["2023-03-01"], "Mar"}, {curdate["2023-04-01"], "Apr"},
    {curdate["2023-05-01"], "May"}, {curdate["2023-06-01"], "June"}};
```

To adjust to the population size of the province:

```
In[∘]:= popsize = CanadaPop[[Position[CanadaPop[[All, 1]], "AB"][[1, 1]]]]

Out[∘]= {AB, 4 543 111, 439 109}
```

The reported cases among those under and over 70+ will be reported per 100,000 for better comparison among provinces:

```
In[∘]:= per100000Y = 100 000 / (popsize[[2]] - popsize[[3]])
        per100000A = 100 000 / popsize[[3]]
```

Out[∘]= $\frac{50\,000}{2\,052\,001}$

Out[∘]= $\frac{100\,000}{439\,109}$

```
In[∘]:= tabdates = {"2021-12-01", "2021-12-02", "2021-12-03",
    "2021-12-04", "2021-12-05", "2021-12-06", "2021-12-07",
    "2021-12-08", "2021-12-09", "2021-12-10", "2021-12-11", "2021-12-12",
    "2021-12-13", "2021-12-14", "2021-12-15", "2021-12-16", "2021-12-17",
    "2021-12-18", "2021-12-19", "2021-12-20", "2021-12-21", "2021-12-22",
    "2021-12-23", "2021-12-24", "2021-12-25", "2021-12-26", "2021-12-27",
    "2021-12-28", "2021-12-29", "2021-12-30", "2021-12-31", "2022-01-01",
    "2022-01-02", "2022-01-03", "2022-01-04", "2022-01-05", "2022-01-06",
    "2022-01-07", "2022-01-08", "2022-01-09", "2022-01-10", "2022-01-11",
    "2022-01-12", "2022-01-13", "2022-01-14", "2022-01-15", "2022-01-16",
    "2022-01-17", "2022-01-18", "2022-01-19", "2022-01-20", "2022-01-21",
    "2022-01-22", "2022-01-23", "2022-01-24", "2022-01-25", "2022-01-26",
    "2022-01-27", "2022-01-28", "2022-01-29", "2022-01-30", "2022-01-31",
    "2022-02-01", "2022-02-02", "2022-02-03", "2022-02-04", "2022-02-05",
    "2022-02-06", "2022-02-07", "2022-02-08", "2022-02-09", "2022-02-10",
    "2022-02-11", "2022-02-12", "2022-02-13", "2022-02-14", "2022-02-15",
    "2022-02-16", "2022-02-17", "2022-02-18", "2022-02-19", "2022-02-20",
    "2022-02-21", "2022-02-22", "2022-02-23", "2022-02-24", "2022-02-25",
```

```
"2022-02-26", "2022-02-27", "2022-02-28", "2022-03-01", "2022-03-02",
"2022-03-03", "2022-03-04", "2022-03-05", "2022-03-06", "2022-03-07",
"2022-03-08", "2022-03-09", "2022-03-10", "2022-03-11", "2022-03-12",
"2022-03-13", "2022-03-14", "2022-03-15", "2022-03-16", "2022-03-17",
"2022-03-18", "2022-03-19", "2022-03-20", "2022-03-21", "2022-03-22",
"2022-03-23", "2022-03-24", "2022-03-25", "2022-03-26", "2022-03-27",
"2022-03-28", "2022-03-29", "2022-03-30", "2022-03-31", "2022-04-01",
"2022-04-02", "2022-04-03", "2022-04-04", "2022-04-05", "2022-04-06",
"2022-04-07", "2022-04-08", "2022-04-09", "2022-04-10", "2022-04-11",
"2022-04-12", "2022-04-13", "2022-04-14", "2022-04-15", "2022-04-16",
"2022-04-17", "2022-04-18", "2022-04-19", "2022-04-20", "2022-04-21",
"2022-04-22", "2022-04-23", "2022-04-24", "2022-04-25", "2022-04-26",
"2022-04-27", "2022-04-28", "2022-04-29", "2022-04-30", "2022-05-01",
"2022-05-02", "2022-05-03", "2022-05-04", "2022-05-05", "2022-05-06",
"2022-05-07", "2022-05-08", "2022-05-09", "2022-05-10", "2022-05-11",
"2022-05-12", "2022-05-13", "2022-05-14", "2022-05-15", "2022-05-16",
"2022-05-17", "2022-05-18", "2022-05-19", "2022-05-20", "2022-05-21",
"2022-05-22", "2022-05-23", "2022-05-24", "2022-05-25", "2022-05-26",
"2022-05-27", "2022-05-28", "2022-05-29", "2022-05-30", "2022-05-31",
"2022-06-01", "2022-06-02", "2022-06-03", "2022-06-04", "2022-06-05",
"2022-06-06", "2022-06-07", "2022-06-08", "2022-06-09", "2022-06-10",
"2022-06-11", "2022-06-12", "2022-06-13", "2022-06-14", "2022-06-15",
"2022-06-16", "2022-06-17", "2022-06-18", "2022-06-19", "2022-06-20",
"2022-06-21", "2022-06-22", "2022-06-23", "2022-06-24", "2022-06-25",
"2022-06-26", "2022-06-27", "2022-06-28", "2022-06-29", "2022-06-30",
"2022-07-01", "2022-07-02", "2022-07-03", "2022-07-04", "2022-07-05",
"2022-07-06", "2022-07-07", "2022-07-08", "2022-07-09", "2022-07-10",
"2022-07-11", "2022-07-12", "2022-07-13", "2022-07-14", "2022-07-15",
"2022-07-16", "2022-07-17", "2022-07-18", "2022-07-19", "2022-07-20",
"2022-07-21", "2022-07-22", "2022-07-23", "2022-07-24", "2022-07-25",
"2022-07-26", "2022-07-27", "2022-07-28", "2022-07-29", "2022-07-30",
"2022-07-31", "2022-08-01", "2022-08-02", "2022-08-03", "2022-08-04",
"2022-08-05", "2022-08-06", "2022-08-07", "2022-08-08", "2022-08-09",
"2022-08-10", "2022-08-11", "2022-08-12", "2022-08-13", "2022-08-14",
"2022-08-15", "2022-08-16", "2022-08-17", "2022-08-18", "2022-08-19",
"2022-08-20", "2022-08-21", "2022-08-22", "2022-08-23", "2022-08-24",
"2022-08-25", "2022-08-26", "2022-08-27", "2022-08-28", "2022-08-29",
"2022-08-30", "2022-08-31", "2022-09-01", "2022-09-02", "2022-09-03",
"2022-09-04", "2022-09-05", "2022-09-06", "2022-09-07", "2022-09-08",
"2022-09-09", "2022-09-10", "2022-09-11", "2022-09-12", "2022-09-13",
"2022-09-14", "2022-09-15", "2022-09-16", "2022-09-17", "2022-09-18",
"2022-09-19", "2022-09-20", "2022-09-21", "2022-09-22", "2022-09-23",
"2022-09-24", "2022-09-25", "2022-09-26", "2022-09-27", "2022-09-28",
"2022-09-29", "2022-09-30", "2022-10-01", "2022-10-02", "2022-10-03",
```

```
"2022-10-04", "2022-10-05", "2022-10-06", "2022-10-07", "2022-10-08",
"2022-10-09", "2022-10-10", "2022-10-11", "2022-10-12", "2022-10-13",
"2022-10-14", "2022-10-15", "2022-10-16", "2022-10-17", "2022-10-18",
"2022-10-19", "2022-10-20", "2022-10-21", "2022-10-22", "2022-10-23",
"2022-10-24", "2022-10-25", "2022-10-26", "2022-10-27", "2022-10-28",
"2022-10-29", "2022-10-30", "2022-10-31", "2022-11-01", "2022-11-02",
"2022-11-03", "2022-11-04", "2022-11-05", "2022-11-06", "2022-11-07",
"2022-11-08", "2022-11-09", "2022-11-10", "2022-11-11", "2022-11-12",
"2022-11-13", "2022-11-14", "2022-11-15", "2022-11-16", "2022-11-17",
"2022-11-18", "2022-11-19", "2022-11-20", "2022-11-21", "2022-11-22",
"2022-11-23", "2022-11-24", "2022-11-25", "2022-11-26", "2022-11-27",
"2022-11-28", "2022-11-29", "2022-11-30", "2022-12-01", "2022-12-02",
"2022-12-03", "2022-12-04", "2022-12-05", "2022-12-06", "2022-12-07",
"2022-12-08", "2022-12-09", "2022-12-10", "2022-12-11", "2022-12-12",
"2022-12-13", "2022-12-14", "2022-12-15", "2022-12-16", "2022-12-17",
"2022-12-18", "2022-12-19", "2022-12-20", "2022-12-21", "2022-12-22",
"2022-12-23", "2022-12-24", "2022-12-25", "2022-12-26", "2022-12-27",
"2022-12-28", "2022-12-29", "2022-12-30", "2022-12-31", "2023-01-01",
"2023-01-02", "2023-01-03", "2023-01-04", "2023-01-05", "2023-01-06",
"2023-01-07", "2023-01-08", "2023-01-09", "2023-01-10", "2023-01-11",
"2023-01-12", "2023-01-13", "2023-01-14", "2023-01-15", "2023-01-16",
"2023-01-17", "2023-01-18", "2023-01-19", "2023-01-20", "2023-01-21",
"2023-01-22", "2023-01-23", "2023-01-24", "2023-01-25", "2023-01-26",
"2023-01-27", "2023-01-28", "2023-01-29", "2023-01-30", "2023-01-31",
"2023-02-01", "2023-02-02", "2023-02-03", "2023-02-04", "2023-02-05",
"2023-02-06", "2023-02-07", "2023-02-08", "2023-02-09", "2023-02-10",
"2023-02-11", "2023-02-12", "2023-02-13", "2023-02-14", "2023-02-15",
"2023-02-16", "2023-02-17", "2023-02-18", "2023-02-19", "2023-02-20",
"2023-02-21", "2023-02-22", "2023-02-23", "2023-02-24", "2023-02-25",
"2023-02-26", "2023-02-27", "2023-02-28", "2023-03-01", "2023-03-02",
"2023-03-03", "2023-03-04", "2023-03-05", "2023-03-06", "2023-03-07",
"2023-03-08", "2023-03-09", "2023-03-10", "2023-03-11", "2023-03-12",
"2023-03-13", "2023-03-14", "2023-03-15", "2023-03-16", "2023-03-17",
"2023-03-18", "2023-03-19", "2023-03-20", "2023-03-21", "2023-03-22",
"2023-03-23", "2023-03-24", "2023-03-25", "2023-03-26", "2023-03-27",
"2023-03-28", "2023-03-29", "2023-03-30", "2023-03-31", "2023-04-01",
"2023-04-02", "2023-04-03", "2023-04-04", "2023-04-05", "2023-04-06",
"2023-04-07", "2023-04-08", "2023-04-09", "2023-04-10", "2023-04-11",
"2023-04-12", "2023-04-13", "2023-04-14", "2023-04-15", "2023-04-16",
"2023-04-17", "2023-04-18", "2023-04-19", "2023-04-20", "2023-04-21",
"2023-04-22", "2023-04-23", "2023-04-24", "2023-04-25", "2023-04-26",
"2023-04-27", "2023-04-28", "2023-04-29", "2023-04-30", "2023-05-01",
"2023-05-02", "2023-05-03", "2023-05-04", "2023-05-05", "2023-05-06",
"2023-05-07", "2023-05-08", "2023-05-09", "2023-05-10", "2023-05-11",
```

```
"2023-05-12", "2023-05-13", "2023-05-14", "2023-05-15", "2023-05-16",
"2023-05-17", "2023-05-18", "2023-05-19", "2023-05-20", "2023-05-21",
"2023-05-22", "2023-05-23", "2023-05-24", "2023-05-25", "2023-05-26",
"2023-05-27", "2023-05-28", "2023-05-29", "2023-05-30", "2023-05-31"};
```

tab = Table[datetabAB[i], {i, 1, Length[dates]}] // N

```
In[◦]:= tab = {{8.552627411000287`, 2.9605405491574985`},
  {7.626701936305099`, 6.604282763505189`}, {7.967832374350695`,
   4.554677767934613`}, {5.750484527054324`, 2.5050727723640374`},
  {4.239764015709544`, 2.2773388839673063`}, {5.482453468589927`,
   2.0496049955705757`}, {9.08868952792908`, 2.5050727723640374`},
  {8.23586343281509`, 3.6437422143476903`}, {7.919099454629896`,
   4.3269438795378825`}, {7.139372739097106`, 4.3269438795378825`},
  {6.06724850523952`, 2.2773388839673063`}, {4.653993833336338`,
   2.2773388839673063`}, {6.505844782726714`, 2.0496049955705757`},
  {10.52631065969266`, 2.732806660760768`}, {10.989273397040256`,
   3.6437422143476903`}, {12.93859018587223`, 3.4160083259509597`},
  {14.668608835960606`, 3.871476102744421`}, {16.76412438395498`,
   5.010145544728075`}, {13.377186463359424`, 3.188274437554229`},
  {18.737807632647353`, 4.782411656331344`}, {32.13936055586718`,
   7.970686093885573`}, {37.7923792434799`, 5.010145544728075`},
  {46.85670231154858`, 10.47575886624961`}, {58.674435343842426`,
   9.564823312662687`}, {37.4999817251551`, 7.287484428695381`},
  {16.130596427584585`, 5.693347209918266`}, {32.69978913265637`,
   7.970686093885573`}, {65.69197578363753`, 16.624573852961337`},
  {94.32256611960716`, 22.317921062879606`}, {102.75336123130545`,
   28.92220382638479`}, {102.04673389535385`, 33.93234937111287`},
  {75.2679945087746`, 24.59525994684691`}, {44.95611844243741`,
   25.506195500433833`}, {68.2260876091191`, 30.74407493355864`},
  {109.52723707249655`, 42.13076935339517`}, {112.11008181769891`,
   34.387817147906325`}, {144.0545106946829`, 48.50731822850363`},
  {143.7377467164977`, 48.73505211690036`}, {144.12761007426408`,
   53.28972988483497`}, {124.24457882817796`, 48.051850451710166`},
  {107.60228674352499`, 46.913181009726515`}, {158.8205853700851`,
   55.33933488040555`}, {135.5749826632638`, 64.22095652787804`},
  {141.66659762836372`, 94.50956368464323`}, {140.47264109520415`,
   73.78577984054073`}, {122.44146079850839`, 72.8748442869538`},
  {91.0574604983136`, 89.72715202831188`}, {79.77578958294855`,
   54.42839932681863`}, {86.81769648260405`, 46.913181009726515`},
  {81.99313743024491`, 59.89401264834016`}, {83.67442316061249`,
   56.02253654559574`}, {86.91516232204565`, 69.45883596100285`},
  {57.821609248728436`, 55.33933488040555`}, {36.98828606808671`,
   38.25929325065075`}, {64.54675217019874`, 42.58623713018863`},
  {75.0974292897518`, 58.299875429563045`}, {71.49119323041266`,
```

```
 67.18149707703554`}, {68.4210192880023`, 60.577214313530355`},
{66.00873976182272`, 53.972931550025166`}, {39.37619913440588`,
 47.14091489812324`}, {38.377174280129495`, 35.981954366683446`},
{45.3459818002038`, 34.61555103630306`}, {68.59158450702509`,
 49.873721558884014`}, {54.775801766178475`, 51.012391000867666`},
{46.393739574200985`, 42.13076935339517`}, {43.64032960997582`,
 52.37879433124805`}, {29.264118292340015`, 39.3979626926344`},
{23.39180146598369`, 36.665156031873636`}, {34.40544132288434`,
 28.239002161194602`}, {39.20563391538308`, 43.269438795378825`},
{31.72513073824038`, 36.4374221434769`}, {29.361584131781612`,
 44.86357601415594`}, {23.781664823750084`, 28.01126827279787`},
{17.44638526004617`, 30.74407493355864`}, {13.425919383080222`,
 18.901912736928644`}, {19.322602669296945`, 14.574968857390761`},
{21.807981575057713`, 33.47688159431941`}, {19.980497085527738`,
 31.88274437554229`}, {16.447360405769782`, 28.239002161194602`},
{14.132546719031813`, 36.209688255080174`}, {9.356720586393477`,
 20.0405821789123`}, {8.820658469464684`, 23.456590504863257`},
{9.06432306806868`, 18.674178848531913`}, {15.399602631772597`,
 27.328066607607678`}, {15.618900770516193`, 21.862453286086144`},
{12.719292047128633`, 30.74407493355864`}, {12.18322993019984`,
 22.545654951276333`}, {9.502919345555874`, 17.0800416297548`},
{7.480503177142701`, 15.713638299374415`}, {10.891807557598655`,
 15.941372187771146`}, {10.428844820251062`, 27.100332719210947`},
{12.11013055061864`, 31.199542710352098`}, {10.038981462484667`,
 18.44644496013518`}, {8.576993870860687`, 16.396839964564606`},
{7.919099454629896`, 16.396839964564606`}, {5.555552848171127`,
 12.753097750216916`}, {9.478552885695475`, 18.901912736928644`},
{10.57504357941346`, 23.91205828165672`}, {12.158863470339439`,
 18.44644496013518`}, {9.429819965974675`, 18.901912736928644`},
{9.234888287091477`, 28.239002161194602`}, {7.309937958119903`,
 21.634719397689413`}, {8.455161571558689`, 20.72378384410249`},
{9.91714916318267`, 15.485904410977684`}, {11.33040383508585`,
 25.733929388830564`}, {11.769000112573044`, 19.357380513722106`},
{12.719292047128633`, 18.901912736928644`}, {11.427869674527448`,
 21.634719397689413`}, {9.697851024439071`, 18.44644496013518`},
{6.164714344681118`, 20.496049955705757`}, {9.89278270332227`,
 17.990977183341723`}, {16.057497048003388`, 25.05072772364037`},
{15.009739274006202`, 28.239002161194602`}, {16.812857303675777`,
 29.37767160317825`}, {14.156913178892212`, 25.05072772364037`},
{10.112080842065867`, 25.961663277227295`}, {8.601360330721086`,
 24.13979217005345`}, {11.622801353410647`, 21.17925162089595`},
{17.714416318510565`, 33.249147705922674`}, {17.20272066144217`,
 36.665156031873636`}, {17.982447376974964`, 38.94249491584094`},
{16.983422522698575`, 36.89288992027036`}, {12.451260988664236`,
```

```
 34.84328492469979`}, {9.965882082903468`, 25.278461612037102`},
{15.740733069818193`, 40.308898246221325`}, {19.907397705946536`,
 48.73505211690036`}, {20.004863545388137`, 43.04170490698209`},
{17.300186500883772`, 47.596382674916704`}, {16.959056062838176`,
 53.74519766162843`}, {14.44931069721701`, 52.15106044285132`},
{10.501944199832261`, 36.4374221434769`}, {15.570167850795395`,
 45.54677767934613`}, {18.90837285167015`, 53.74519766162843`},
{18.518509493903757`, 60.12174653673689`}, {18.78654055236815`,
 57.38893987597613`}, {14.181279638752613`, 60.12174653673689`},
{12.426894528803835`, 51.92332655445459`}, {11.111105696342253`,
 41.21983379980825`}, {15.667633690236993`, 38.48702713904748`},
{21.27191945812892`, 57.16120598757939`}, {22.295310772265704`,
 71.2807070681767`}, {18.689074712926555`, 61.71588375551401`},
{17.787515698091767`, 54.20066543842189`}, {14.37621131763581`,
 56.478004322389204`}, {12.012664711177042`, 41.90303546499844`},
{13.64521752182382`, 31.88274437554229`}, {19.615000187621742`,
 62.17135153230747`}, {16.520459785350983`, 60.12174653673689`},
{16.349894566328185`, 56.25027043399247`}, {13.62085106196342`,
 44.86357601415594`}, {11.695900732991845`, 56.02253654559574`},
{10.380111900530263`, 41.67530157660171`}, {12.889857266151429`,
 32.33821215233575`}, {13.888882120427816`, 42.3585032417919`},
{14.254379018333811`, 42.58623713018863`}, {13.937615040148616`,
 46.68544712132979`}, {12.012664711177042`, 44.86357601415594`},
{9.649118104718273`, 44.18037434896575`}, {2.192981387435971`,
 10.02029108945615`}, {15.1315715733082`, 45.54677767934613`},
{12.475627448524635`, 32.11047826393902`}, {10.550677119553061`,
 34.61555103630306`}, {10.916174017459056`, 35.07101881309652`},
{8.576993870860687`, 26.872598830814216`}, {8.35769573211709`,
 24.36752605845018`}, {6.335279563703916`, 21.40698550929268`},
{9.11305598778948`, 23.91205828165672`}, {9.551652265276674`,
 35.07101881309652`}, {9.015590148347881`, 22.545654951276333`},
{8.601360330721086`, 31.65501048714556`}, {6.725142921470312`,
 17.990977183341723`}, {5.28752178970673`, 16.852307741358068`},
{4.385962774871942`, 11.614428308233263`}, {6.189080804541518`,
 15.030436634184223`}, {7.919099454629896`, 14.574968857390761`},
{8.576993870860687`, 23.001122728069795`}, {7.163739198957505`,
 15.713638299374415`}, {7.066273359515907`, 11.614428308233263`},
{4.824559052359136`, 13.89176719220057`}, {1.3401552923219824`,
 7.515218317092112`}, {9.210521827231078`, 13.664033303803839`},
{6.8713416806327094`, 17.0800416297548`}, {5.3606211692879295`,
 11.614428308233263`}, {5.726118067193925`, 14.347234968994032`},
{4.653993833336338`, 17.0800416297548`}, {3.923000037524348`,
 7.742952205488842`}, {3.728068358641151`, 7.742952205488842`},
{4.775826132638337`, 8.198419982282303`}, {5.823583906635523`,
```

```
11.614428308233263`}, {5.190055950265132`, 11.386694419836532`},
{4.58089445375514`, 13.208565527010379`}, {3.6549689790599516`,
 7.515218317092112`}, {2.802142883945963`, 6.148814986711727`},
{2.8996087233875616`, 6.148814986711727`}, {3.192006241712358`,
 7.287484428695381`}, {4.191031095988745`, 7.742952205488842`},
{3.362571460735156`, 10.47575886624961`}, {2.875242263527162`,
 6.376548875108458`}, {3.192006241712358`, 10.248024977852879`},
{2.8508758036667623`, 10.248024977852879`}, {1.0964906937179855`,
 4.782411656331344`}, {2.729043504364764`, 5.921081098314997`},
{3.5087702198975537`, 7.970686093885573`}, {3.0214410226895603`,
 6.604282763505189`}, {2.26608076701717`, 7.742952205488842`},
{2.4610124459003675`, 7.0597505402986505`}, {2.26608076701717`,
 6.376548875108458`}, {1.656919270507178`, 5.465613321521536`},
{2.631577664923165`, 4.554677767934613`}, {3.0214410226895603`,
 9.564823312662687`}, {3.2651056212935567`, 9.109355535869225`},
{3.728068358641151`, 10.02029108945615`}, {2.7777764240855634`,
 9.564823312662687`}, {2.2417143071567702`, 7.515218317092112`},
{1.8518509493903756`, 8.653887759075765`}, {3.240739161433157`,
 10.70349275464634`}, {4.922024891800735`, 10.248024977852879`},
{3.8255341980827495`, 7.0597505402986505`}, {4.166664636128345`,
 13.436299415407108`}, {3.7037018987807513`, 8.653887759075765`},
{3.849900657943149`, 7.515218317092112`}, {2.729043504364764`,
 4.3269438795378825`}, {3.6062360593391523`, 6.376548875108458`},
{5.190055950265132`, 8.426153870679034`}, {5.458087008729528`,
 9.109355535869225`}, {4.5077950741739405`, 11.386694419836532`},
{4.848925512219536`, 10.47575886624961`}, {3.8255341980827495`,
 10.931226643043072`}, {3.484403760037154`, 8.198419982282303`},
{6.432745403145515`, 8.881621647472496`}, {6.140347884820719`,
 15.485904410977684`}, {4.824559052359136`, 15.030436634184223`},
{5.872316826356323`, 7.742952205488842`}, {3.679335438920351`,
 15.030436634184223`}, {4.288496935430343`, 12.753097750216916`},
{2.655944124783565`, 12.069896085026725`}, {4.264130475569944`,
 9.564823312662687`}, {5.3606211692879295`, 17.53550940654826`},
{4.093565256547146`, 15.713638299374415`}, {4.385962774871942`,
 16.396839964564606`}, {5.28752178970673`, 20.72378384410249`},
{3.6062360593391523`, 11.386694419836532`}, {2.363546606458769`,
 12.069896085026725`}, {3.5087702198975537`, 15.485904410977684`},
{4.215397555849144`, 16.624573852961337`}, {5.701751607333525`,
 22.317921062879606`}, {4.361596315011543`, 19.812848290515568`},
{4.361596315011543`, 17.763243294944992`}, {4.093565256547146`,
 17.53550940654826`}, {3.849900657943149`, 12.297629973423454`},
{3.8255341980827495`, 13.208565527010379`}, {4.970757811521534`,
 16.624573852961337`}, {3.874267117803549`, 20.0405821789123`},
{4.897658431940335`, 15.941372187771146`}, {4.848925512219536`,
```

17.990977183341723`}, {3.5331366797579533`, 20.0405821789123`},
{1.9493167888319742`, 10.47575886624961`}, {3.2651056212935567`,
15.485904410977684`}, {5.482453468589927`, 20.0405821789123`},
{3.5818695994787526`, 19.129646625325375`}, {3.8986335776639485`,
25.506195500433833`}, {2.802142883945963`, 17.53550940654826`},
{2.4610124459003675`, 20.95151773249922`}, {2.9727081029687605`,
18.901912736928644`}, {2.485378905760767`, 12.525363861820185`},
{3.679335438920351`, 24.82299383524364`}, {3.728068358641151`,
20.0405821789123`}, {3.3382050008747557`, 11.386694419836532`},
{2.729043504364764`, 12.297629973423454`}, {2.729043504364764`,
16.396839964564606`}, {2.3391801465983693`, 15.485904410977684`},
{2.2417143071567702`, 11.158960531439803`}, {2.6803105846439648`,
13.664033303803839`}, {2.6072112050627654`, 20.95151773249922`},
{2.5828447452023657`, 15.485904410977684`}, {2.5828447452023657`,
9.792557201059418`}, {2.1442484677151716`, 11.842162196629994`},
{1.5594534310655794`, 12.297629973423454`}, {1.681285730367578`,
10.248024977852879`}, {2.0955155479943723`, 6.832016651901919`},
{3.362571460735156`, 17.30777551815153`}, {2.7777764240855634`,
13.89176719220057`}, {2.8265093438063627`, 14.119501080597301`},
{1.9493167888319742`, 11.158960531439803`}, {1.7300186500883772`,
12.753097750216916`}, {2.192981387435971`, 11.614428308233263`},
{3.8255341980827495`, 16.169106076167875`}, {3.2163727015727575`,
13.208565527010379`}, {3.411304380455955`, 13.89176719220057`},
{2.5341118254815664`, 14.802702745787492`}, {2.9239751832479612`,
15.941372187771146`}, {1.7056521902279775`, 14.574968857390761`},
{2.3391801465983693`, 23.001122728069795`}, {2.485378905760767`,
16.624573852961337`}, {4.069198796686746`, 20.496049955705757`},
{3.7037018987807513`, 21.634719397689413`}, {2.9970745628291606`,
15.941372187771146`}, {3.094540402270759`, 18.44644496013518`},
{2.119882007854772`, 15.258170522580954`}, {2.7046770445043644`,
14.574968857390761`}, {4.069198796686746`, 23.68432439325999`},
{3.0214410226895603`, 18.674178848531913`}, {3.5331366797579533`,
23.68432439325999`}, {1.9980497085527735`, 19.585114402118837`},
{2.046782628273573`, 18.901912736928644`}, {2.0224161684131734`,
19.585114402118837`}, {2.509745365621167`, 19.585114402118837`},
{3.7768012783619502`, 25.733929388830564`}, {2.412279526179568`,
18.44644496013518`}, {2.8265093438063627`, 21.862453286086144`},
{2.192981387435971`, 20.95151773249922`}, {2.3148136867379696`,
16.624573852961337`}, {1.5350869712051798`, 18.674178848531913`},
{1.8518509493903756`, 13.436299415407108`}, {2.1442484677151716`,
19.129646625325375`}, {2.9239751832479612`, 23.91205828165672`},
{3.5087702198975537`, 25.961663277227295`}, {2.6072112050627654`,
25.733929388830564`}, {2.0955155479943723`, 17.53550940654826`},
{2.1686149275755713`, 21.40698550929268`}, {2.119882007854772`,

```
 16.396839964564606`}, {2.631577664923165`, 28.694469937988064`},
{2.8508758036667623`, 27.328066607607678`}, {2.8996087233875616`,
 26.189397165624026`}, {2.26608076701717`, 24.82299383524364`},
{2.192981387435971`, 15.485904410977684`}, {2.192981387435971`,
 23.001122728069795`}, {2.412279526179568`, 20.95151773249922`},
{3.167639781851958`, 21.862453286086144`}, {2.046782628273573`,
 14.574968857390761`}, {3.630602519199552`, 22.317921062879606`},
{1.7787515698091765`, 16.396839964564606`}, {2.655944124783565`,
 19.357380513722106`}, {1.681285730367578`, 19.585114402118837`},
{2.119882007854772`, 15.713638299374415`}, {3.7768012783619502`,
 24.82299383524364`}, {2.802142883945963`, 17.763243294944992`},
{3.1432733219915585`, 24.36752605845018`}, {2.7046770445043644`,
 20.72378384410249`}, {2.9239751832479612`, 15.941372187771146`},
{1.8274844895299758`, 14.347234968994032`}, {2.4610124459003675`,
 16.852307741358068`}, {2.875242263527162`, 25.278461612037102`},
{2.1686149275755713`, 20.95151773249922`}, {3.484403760037154`,
 22.317921062879606`}, {1.9249503289715746`, 16.169106076167875`},
{2.5341118254815664`, 21.40698550929268`}, {2.0224161684131734`,
 17.53550940654826`}, {2.363546606458769`, 18.21871107173845`},
{4.093565256547146`, 20.95151773249922`}, {3.874267117803549`,
 19.357380513722106`}, {2.558478285341966`, 15.713638299374415`},
{2.3148136867379696`, 13.436299415407108`}, {3.3382050008747557`,
 15.941372187771146`}, {1.6081863507863787`, 14.347234968994032`},
{2.9239751832479612`, 15.258170522580954`}, {2.7046770445043644`,
 15.941372187771146`}, {3.874267117803549`, 26.189397165624026`},
{3.094540402270759`, 17.30777551815153`}, {2.5828447452023657`,
 22.545654951276333`}, {2.3391801465983693`, 20.268316067309026`},
{1.7056521902279775`, 13.89176719220057`}, {1.437621131763581`,
 14.574968857390761`}, {3.167639781851958`, 18.901912736928644`},
{1.7543851099487768`, 18.21871107173845`}, {2.6803105846439648`,
 26.644864942417485`}, {1.900583869111175`, 13.89176719220057`},
{2.1442484677151716`, 18.21871107173845`}, {1.681285730367578`,
 11.614428308233263`}, {1.5594534310655794`, 12.069896085026725`},
{2.655944124783565`, 14.119501080597301`}, {2.046782628273573`,
 18.674178848531913`}, {3.1432733219915585`, 17.0800416297548`},
{2.046782628273573`, 18.674178848531913`}, {1.5594534310655794`,
 8.426153870679034`}, {0.3167639781851958`, 2.9605405491574985`},
{2.7534099642251637`, 20.268316067309026`}, {2.8996087233875616`,
 15.030436634184223`}, {2.0711490881339727`, 13.436299415407108`},
{2.2417143071567702`, 14.574968857390761`}, {1.6081863507863787`,
 10.248024977852879`}, {2.192981387435971`, 11.158960531439803`},
{1.0721242338575858`, 7.970686093885573`}, {1.681285730367578`,
 8.653887759075765`}, {2.412279526179568`, 15.258170522580954`},
{1.583819890925979`, 10.70349275464634`}, {2.046782628273573`,
```

15.485904410977684`}, {1.1695900732991846`, 11.158960531439803`},
{1.6081863507863787`, 10.02029108945615`}, {1.145223613438785`,
7.742952205488842`}, {0.9259254746951878`, 11.386694419836532`},
{1.9249503289715746`, 11.158960531439803`}, {1.9493167888319742`,
10.248024977852879`}, {2.5341118254815664`, 22.090187174482875`},
{1.583819890925979`, 13.436299415407108`}, {1.8518509493903756`,
10.70349275464634`}, {1.437621131763581`, 10.248024977852879`},
{1.437621131763581`, 13.664033303803839`}, {2.26608076701717`,
14.347234968994032`}, {1.583819890925979`, 18.44644496013518`},
{2.6803105846439648`, 16.396839964564606`}, {1.4132546719031813`,
17.30777551815153`}, {1.7300186500883772`, 12.753097750216916`},
{1.3157888324615825`, 15.258170522580954`}, {1.5350869712051798`,
11.614428308233263`}, {2.2173478472963706`, 21.40698550929268`},
{1.3157888324615825`, 18.21871107173845`}, {2.0224161684131734`,
15.941372187771146`}, {1.3888882120427817`, 13.436299415407108`},
{1.437621131763581`, 10.70349275464634`}, {0.9746583944159871`,
8.198419982282303`}, {0.8528260951139888`, 7.970686093885573`},
{1.681285730367578`, 12.069896085026725`}, {1.3401552923219824`,
11.614428308233263`}, {1.5594534310655794`, 10.248024977852879`},
{1.1939565331595843`, 14.347234968994032`}, {1.145223613438785`,
5.693347209918266`}, {0.7797267155327897`, 8.426153870679034`},
{0.7553602556723901`, 4.3269438795378825`}, {1.9736832486923739`,
11.614428308233263`}, {1.3888882120427817`, 5.237879433124805`},
{1.583819890925979`, 11.614428308233263`}, {0.9746583944159871`,
7.0597505402986505`}, {0.9259254746951878`, 10.47575886624961`},
{0.7553602556723901`, 5.465613321521536`}, {0.828459635253589`,
5.010145544728075`}, {1.2670559127407832`, 10.02029108945615`},
{1.5350869712051798`, 7.970686093885573`}, {1.1939565331595843`,
7.742952205488842`}, {1.4619875916239806`, 6.376548875108458`},
{1.0964906937179855`, 5.465613321521536`}, {0.8040931753931894`,
5.693347209918266`}, {0.9259254746951878`, 5.010145544728075`},
{1.0721242338575858`, 6.832016651901919`}, {1.437621131763581`,
7.742952205488842`}, {0.9746583944159871`, 7.970686093885573`},
{0.5604285767891926`, 4.554677767934613`}, {0.6578944162307913`,
3.871476102744421`}, {0.6578944162307913`, 3.871476102744421`},
{0.7797267155327897`, 6.376548875108458`}, {0.8040931753931894`,
7.515218317092112`}, {0.8771925549743884`, 9.109355535869225`},
{0.7066273359515907`, 3.6437422143476903`}, {1.3888882120427817`,
7.28748442869538`}, {0.5604285767891926`, 5.465613321521536`},
{0.7066273359515907`, 3.188274437554229`}, {0.8040931753931894`,
4.782411656331344`}, {0.6335279563703916`, 5.921081098314997`},
{0.609161496509992`, 8.881621647472496`}, {0.7309937958119903`,
5.921081098314997`}, {0.7309937958119903`, 5.237879433124805`},
{0.7797267155327897`, 3.6437422143476903`}, {0.609161496509992`,

```
 6.148814986711727`}, {0.7066273359515907`, 6.604282763505189`},
{0.4873291972079356`, 7.0597505402986505`}, {0.7066273359515907`,
 5.010145544728075`}, {0.6335279563703916`, 3.871476102744421`},
{0.7797267155327897`, 4.0992099911411515`}, {0.7553602556723901`,
 5.465613321521536`}, {0.609161496509992`, 3.4160083259509597`},
{0.8040931753931894`, 5.010145544728075`}, {0.7553602556723901`,
 4.554677767934613`}, {0.4873291972079356`, 5.693347209918266`},
{0.9990248542763868`, 6.148814986711727`}, {0.682260876091191`,
 5.237879433124805`}, {0.9015590148347881`, 7.742952205488842`},
{0.5847950366495923`, 3.6437422143476903`}, {0.682260876091191`,
 7.287484428695381`}, {0.609161496509992`, 5.921081098314997`},
{1.0721242338575858`, 10.931226643043072`}, {0.8528260951139888`,
 3.4160083259509597`}, {0.7309937958119903`, 5.465613321521536`},
{0.5604285767891926`, 6.376548875108458`}, {0.7309937958119903`,
 4.782411656331344`}, {0.682260876091191`, 4.554677767934613`},
{0.9502919345555875`, 6.832016651901919`}, {0.9015590148347881`,
 5.693347209918266`}, {1.0477577739971862`, 5.237879433124805`},
{0.6335279563703916`, 6.832016651901919`}, {0.5604285767891926`,
 5.693347209918266`}, {0.682260876091191`, 5.693347209918266`},
{0.5116956570683933`, 4.3269438795378825`}, {0.7797267155327897`,
 7.0597505402986505`}, {0.5360621169287929`, 4.782411656331344`},
{1.0964906937179855`, 6.376548875108458`}, {0.6335279563703916`,
 6.832016651901919`}, {0.5847950366495923`, 5.465613321521536`},
{0.4873291972079356`, 5.921081098314997`}, {0.5360621169287929`,
 6.376548875108458`}, {0.4629627373475939`, 4.782411656331344`},
{0.682260876091191`, 9.109355535869225`}, {0.5360621169287929`,
 7.0597505402986505`}, {0.38986335776639486`, 6.832016651901919`},
{0.36549689790599516`, 5.921081098314997`}, {0.5604285767891926`,
 4.782411656331344`}, {0.4629627373475939`, 6.148814986711727`},
{0.5360621169287929`, 5.921081098314997`}, {0.4385962774871942`,
 6.604282763505189`}, {0.4142298176267945`, 6.148814986711727`},
{0.5116956570683933`, 4.0992099911411515`}, {0.4385962774871942`,
 4.3269438795378825`}, {0.5604285767891926`, 5.693347209918266`},
{0.5360621169287929`, 5.237879433124805`}, {0.609161496509992`,
 9.337089424265956`}, {0.8771925549743884`, 8.198419982282303`},
{0.7066273359515907`, 6.376548875108458`}, {0.4873291972079356`,
 6.148814986711727`}, {0.5116956570683933`, 6.148814986711727`},
{0.38986335776639486`, 7.287484428695381`}, {0.609161496509992`,
 5.465613321521536`}, {0.5604285767891926`, 6.604282763505189`},
{0.5604285767891926`, 6.376548875108458`}, {0.682260876091191`,
 4.0992099911411515`}, {0.7797267155327897`, 6.148814986711727`},
{0.4629627373475939`, 3.871476102744421`}, {0.36549689790599516`,
 3.871476102744421`}, {0.5360621169287929`, 5.237879433124805`},
{0.5847950366495923`, 6.148814986711727`}, {0.6335279563703916`,
```

```
 5.465613321521536`}, {0.5360621169287929`, 5.921081098314997`},
{0.36549689790599516`, 7.970686093885573`}, {0.24366459860399678`,
 6.832016651901919`}, {0.24366459860399678`, 4.3269438795378825`},
{0.48732919720799356`, 7.0597505402986505`}, {0.609161496509992`,
 5.465613321521536`}, {0.48732919720799356`, 4.554677767934613`},
{0.29239751832479616`, 7.0597505402986505`}, {0.5360621169287929`,
 5.237879433124805`}, {0.4142298176267945`, 5.465613321521536`},
{0.3167639781851958`, 4.3269438795378825`}, {0.3167639781851958`,
 3.4160083259509597`}, {0.5360621169287929`, 6.148814986711727`},
{0.36549689790599516`, 3.4160083259509597`}, {0.29239751832479616`,
 4.782411656331344`}, {0.36549689790599516`, 3.188274437554229`},
{0.2192981387435971`, 4.0992099911411515`}, {0.26803105846439645`,
 4.782411656331344`}, {0.17056521902279775`, 3.871476102744421`},
{0.48732919720799356`, 4.782411656331344`}, {0.26803105846439645`,
 2.2773388839673063`}, {0.4142298176267945`, 5.237879433124805`},
{0.2192981387435971`, 2.5050727723640374`}, {0.2192981387435971`,
 5.465613321521536`}, {0.36549689790599516`, 2.5050727723640374`},
{0.26803105846439645`, 2.732806660760768`}, {0.5360621169287929`,
 3.871476102744421`}, {0.3167639781851958`, 3.871476102744421`}};
```

Enter the following to avoid upload:

```
In[∘]:= dates = tabdates;
```

```
In[∘]:= For[i = 1, i <= Length[dates], i++, datetabAB[i] = tab[[i]]]
```

Overall Analysis by 70+

Cubic Spline fit (https://mathematica.stackexchange.com/questions/33206/implementation-of-smoothing-splines-function)

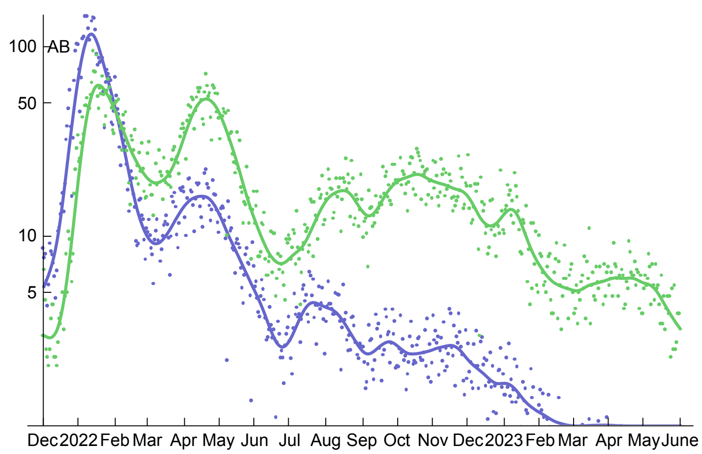
```
In[∘]:= CubicSplSmooth[data_, lambda_] := Module[{M, Knots, X, Dsq, a}, M = Length@data;
    Knots = Flatten@{1, 1, 1, Range@M, M, M, M};
    X = Table[Evaluate@N@BSplineBasis[{3, Knots}, n, t], {t, 1, M}, {n, 0, M + 1}];
    Dsq = Differences[X, 2];
    a = LinearSolve[Transpose[X].X + lambda * Transpose[Dsq].Dsq,
      Transpose[X].data, Method → "Multifrontal"];
    Return[
     X.
      a]];
```

Cubic spline fit to all dates (Y is younger, A is 70+), using lambda = 3 to provide a good fit but not overfit to the data:

In[∘]:=
```
modYAB = CubicSplSmooth[Table[{i, Log[Max[1, datetabAB[i][[1]]]]},
    {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
modAAB = CubicSplSmooth[Table[{i, Log[Max[1, datetabAB[i][[2]]]]},
    {i, 1, Length[dates]}], 10^lambda /. lambda → 3];
```

In[∘]:=
```
plotAB = Show[
  ListPlot[Table[{i, Log[datetabAB[i][[1]]]}, {i, 1, Length[dates]}],
   PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.4, 0.8]}],
  ListLinePlot[modYAB, PlotStyle → RGBColor[0.4, 0.4, 0.8], PlotRange → All],
  ListLinePlot[modAAB, PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
  ListPlot[Table[{i, Log[datetabAB[i][[2]]]}, {i, 1, Length[dates]}],
   PlotRange → {Automatic, {0, 10}}, PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
  Graphics[Text["AB", {5, 4.6}, {-1, 0}]],
  PlotRange → {Automatic, {0, 5}}, AxesOrigin → {1, 0},
  Ticks → {xaxis, {{Log[5], "5"}, {Log[10], "10"}, {Log[50], "50"}, {Log[100], "100"},
     {Log[500], "500"}, {Log[1000], "1000"}, {Log[5000], "5000"}}}
 ]
```

Out[∘]=

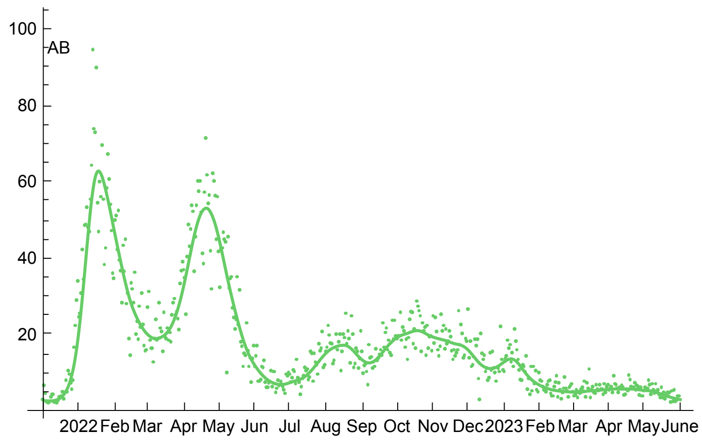


The above illustrates the major drop off in reported cases among those <70:

70+ on a non-log plot

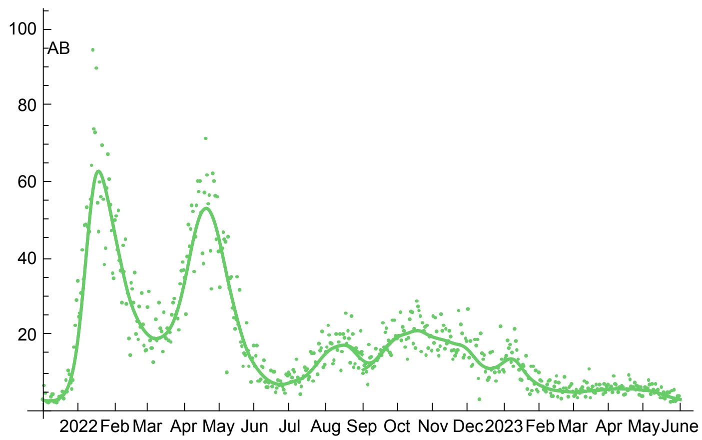
In[◦]:=
```
Show[
 ListLinePlot[Table[{modAAB[[i, 1]], Exp[modAAB[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabAB[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["AB", {5, 95}, {-1, 0}]],
 PlotRange → {Automatic, {0, 100}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

Out[◦]=



In[◦]:=
```
modAAB1 = CubicSplSmooth[Table[{i, Log[Max[1, datetabAB[i][[2]]]]},
    {i, 1, Position[dates, myenddate1][[1, 1]]}], 10^lambda /. lambda → 3];
modAAB2 = CubicSplSmooth[Table[{i, Log[Max[1, datetabAB[i][[2]]]]},
    {i, Position[dates, mystartdate2][[1, 1]], Length[dates]}],
   10^lambda /. lambda → 3];
```

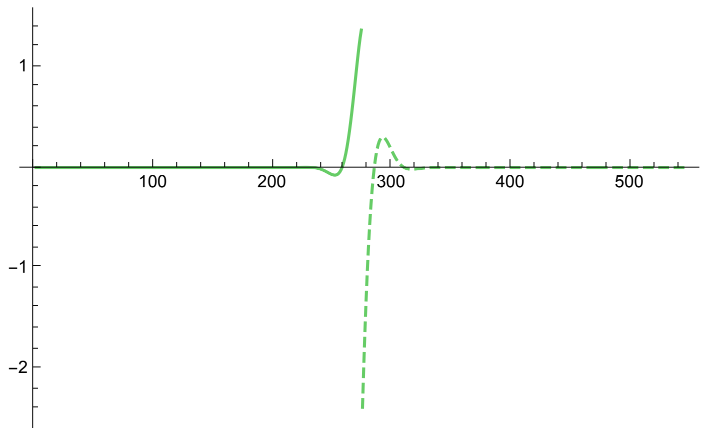
In[∘]:=
```
Show[
 ListLinePlot[Table[{modAAB[[i, 1]], Exp[modAAB[[i, 2]]]}, {i, 1, Length[dates]}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAAB1[[i, 1]], Exp[modAAB1[[i, 2]]]},
   {i, 1, Length[modAAB1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAAB2[[i, 1]], Exp[modAAB2[[i, 2]]]},
   {i, 1, Length[modAAB2]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabAB[i][[2]]}, {i, 1, Length[dates]}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 Graphics[Text["AB", {5, 95}, {-1, 0}]],
 PlotRange → {Automatic, {0, 100}}, Ticks → {xaxis, Automatic}, AxesOrigin → {1, 0}
]
```

Out[∘]=


The fit is very similar, with a slight discrepancy in numbers at the joint, so we just use the overall spline fit for the analysis:

In[◦]:=
```
Show[
 ListLinePlot[Table[{modAAB1[[i, 1]], Exp[modAAB1[[i, 2]]] - Exp[modAAB[[i, 2]]]},
   {i, 1, Length[modAAB1]}], PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListLinePlot[Table[{modAAB2[[i, 1]], Exp[modAAB2[[i, 2]]] - Exp[modAAB[[
       i + Position[dates, myenddate1][[1, 1]], 2]]]}, {i, 1, Length[modAAB2]}],
  PlotStyle → {Dashed, RGBColor[0.4, 0.8, 0.4]}, PlotRange → All],
 PlotRange → All
]
```

Out[◦]=



Merging with Omicron subvariant frequencies

In[◦]:=
```
graphheight = 95;
```

In[◦]:=
```
colourBA1 = RGBColor[0.5, 0.1, 0.1];
colourBA11 = RGBColor[0.75, 0.1, 0.1];
colourBA2 = RGBColor[1, 0.1, 0.1];
colourBA5 = RGBColor[1, 0.5, 0.1];
colourBA45 = colourBA5;
colourBA4 = colourBA5;
colourBQ = RGBColor[1, 0.75, 0.5];
colourXBB15 = RGBColor[0.2, 0.5, 0.8];
colourXBBO = RGBColor[0.5, 0.7, 0.8];

{colourBA1, colourBA11, colourBA2, colourBA45, colourBQ, colourXBB15, colourXBBO}
```

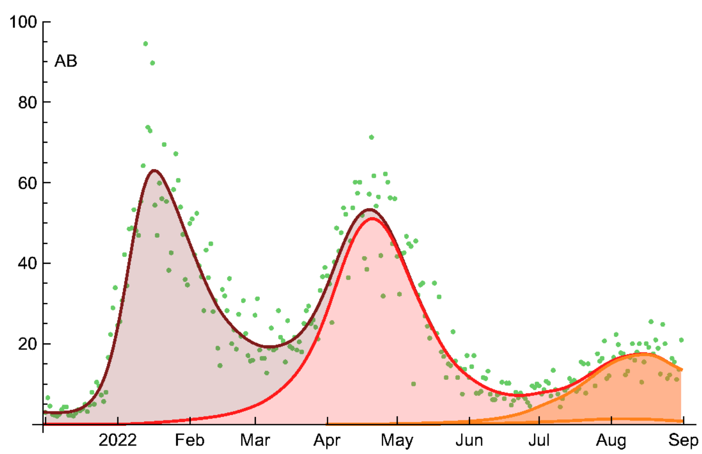
Out[◦]= 

From VariantData_AlbertaOmicron.nb (BA.5 stands for BA.4 and BA.5 and the initial frequency is at "2022-04-1" based on a separate analysis):

In[◦]:=
```
ABML = {pBA2 → 0.00011225159144028347`, sBA2 → 0.0867477034172624`,
   pBA4 → 0.000063200803465215 72`, pBA5 → 0.00011548973437613636`,
   sBA4 → 0.08609383566833843`, sBA5 → 0.10006163455695519`};
```

In[◦]:=
```
endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modAAB[[i, 1]],
    1 * Exp[modAAB[[i, 2]]] /. t -> modAAB[[i, 1]] /. ABML}, {i, 1, endtime}];
tabBA2 = Table[{modAAB[[i, 1]], pBA2 Exp[sBA2 t] / ((1 - pBA2) + pBA2 Exp[sBA2 t]) * Exp[modAAB[[i, 2]]] /.
      t -> modAAB[[i, 1]] /. ABML}, {i, 1, endtime}];
tabBA4 = Table[{modAAB[[i, 1]], pBA4 Exp[sBA4 t] / ((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
       Exp[modAAB[[i, 2]]] /. t -> modAAB[[i, 1]] - curdate["2022-03-31"] /.
     ABML}, {i, curdate["2022-04-1"], endtime}];
tabBA5 = Table[{modAAB[[i, 1]], (pBA5 Exp[sBA5 t] + pBA4 Exp[sBA4 t]) / ((1 - pBA4 - pBA5) + pBA4 Exp[sBA4 t] + pBA5 Exp[sBA5 t]) *
       Exp[modAAB[[i, 2]]] /. t -> modAAB[[i, 1]] - curdate["2022-03-31"] /.
     ABML}, {i, curdate["2022-04-1"], endtime}];

Show[
 ListLinePlot[Table[{modAAB[[i, 1]], Exp[modAAB[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabAB[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA2, tabBA4, tabBA5},
  PlotStyle → {colourBA1, colourBA2, colourBA4, colourBA5}, PlotRange → All,
      Filling → {1 → {2}, 2 → Axis, 3 → {4}, 4 → Axis}],
 Graphics[Text["AB", {5, 0.95 * graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[◦]=

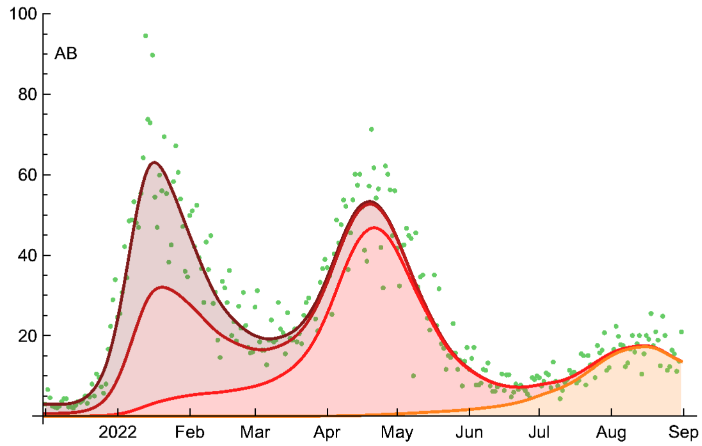


From duotang/VirusSeq (see code above): mutantNames = list(“BA.1.1*”, “BA.2*”, “BA.4/5*”, “BA.1*”)

```
In[•]:= ABML1
Out[•]= {0.13608, 0.845212, 0.00324064, 0.0260576, 0.0712305, 0.151729}

In[•]:= dataname = ABML1;
        refdate = "2022-04-15"; (* t=0 date used in duotang code above *)
        ABML = {pBA11 → dataname[[1]], pBA2 → dataname[[2]], pBA45 → dataname[[3]],
          sBA11 → dataname[[4]], sBA2 → dataname[[5]], sBA45 → dataname[[6]]}
Out[•]= {pBA11 → 0.13608, pBA2 → 0.845212, pBA45 → 0.00324064,
         sBA11 → 0.0260576, sBA2 → 0.0712305, sBA45 → 0.151729}
```

```
In[∘]:= endtime = Length[dates];
endtime = Position[dates, "2022-08-31"][[1, 1]];

tabBA1 = Table[{modAAB[[i, 1]],
    1 * Exp[modAAB[[i, 2]]] /. t -> modAAB[[i, 1]] /. ABML}, {i, 1, endtime}];
tabBA11 = Table[{modAAB[[i, 1]],
     (pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
     ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAAB[[i, 2]]] /.
     t -> modAAB[[i, 1]] - curdate[refdate] /. ABML}, {i, 1, endtime}];
tabBA2 = Table[{modAAB[[i, 1]],
     (pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) /
     ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAAB[[i, 2]]] /.
     t -> modAAB[[i, 1]] - curdate[refdate] /. ABML}, {i, 1, endtime}];
tabBA45 = Table[{modAAB[[i, 1]],
     (pBA45 Exp[sBA45 t]) /
     ((1 - pBA11 - pBA2 - pBA45) + pBA11 Exp[sBA11 t] + pBA2 Exp[sBA2 t] + pBA45 Exp[sBA45 t]) *
      Exp[modAAB[[i, 2]]] /.
     t -> modAAB[[i, 1]] - curdate[refdate] /. ABML}, {i, 1, endtime}];

Show[
 ListLinePlot[Table[{modAAB[[i, 1]], Exp[modAAB[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabAB[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45}, PlotStyle →
   {colourBA1, colourBA11, colourBA2, colourBA4, colourBA5}, PlotRange → All,
     Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis}],
 Graphics[Text["AB", {5, 0.95 * graphheight}, {-1, 0}]],
 PlotRange → {{1, endtime + 1}, {0, graphheight}}, Ticks → {xaxis, Automatic}
]
```

Out[∘]=



From duotang/VirusSeq (see code above): mutantNames = list("XBB.1.5*", "XBB.Other*", "BQ*", "BA.5*")

In[∘]:=
```
ABML2
```

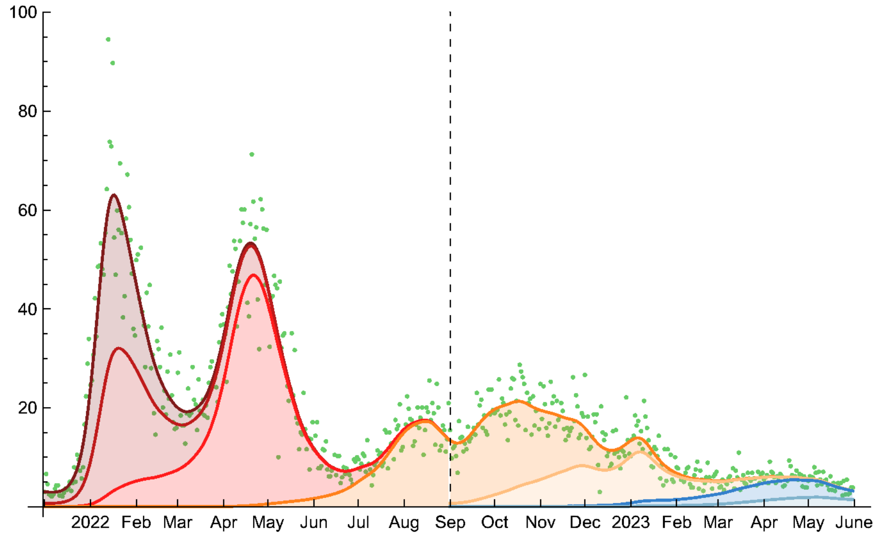
Out[∘]=
```
{0.0911895, 0.0121622, 0.735102, 0.074224, 0.0873077, 0.033236}
```

In[∘]:=
```
dataname = ABML2;
refdate = "2023-01-14"; (* t=0 date used in duotang code above *)
ABML = {pXBB15 → dataname[[1]], pXBBO → dataname[[2]], pBQ → dataname[[3]],
  sXBB15 → dataname[[4]], sXBBO → dataname[[5]], sBQ → dataname[[6]]}
```

Out[∘]=
```
{pXBB15 → 0.0911895, pXBBO → 0.0121622, pBQ → 0.735102,
 sXBB15 → 0.074224, sXBBO → 0.0873077, sBQ → 0.033236}
```

```
In[•]:= starttime = Position[dates, "2022-09-01"][[1, 1]];
endtime = Length[dates];

tabBA5 = Table[{modAAB[[i, 1]], 1 * Exp[modAAB[[i, 2]]] /. t -> modAAB[[i, 1]] /. ABML},
    {i, starttime, endtime}];
tabBQ = Table[{modAAB[[i, 1]],
      (pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
      ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
        Exp[modAAB[[i, 2]]] /.
       t -> modAAB[[i, 1]] - curdate[refdate] /. ABML}, {i, starttime, endtime}];
tabXBB15 = Table[{modAAB[[i, 1]],
      (pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) /
      ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
        Exp[modAAB[[i, 2]]] /.
       t -> modAAB[[i, 1]] - curdate[refdate] /. ABML}, {i, starttime, endtime}];
tabXBBO = Table[{modAAB[[i, 1]],
      (pXBBO Exp[sXBBO t]) /
      ((1 - pBQ - pXBB15 - pXBBO) + pBQ Exp[sBQ t] + pXBB15 Exp[sXBB15 t] + pXBBO Exp[sXBBO t]) *
        Exp[modAAB[[i, 2]]] /.
       t -> modAAB[[i, 1]] - curdate[refdate] /. ABML}, {i, starttime, endtime}];

Show[
 ListLinePlot[Table[{modAAB[[i, 1]], Exp[modAAB[[i, 2]]]}, {i, 1, endtime}],
  PlotStyle → RGBColor[0.4, 0.8, 0.4], PlotRange → All],
 ListPlot[Table[{i, datetabAB[i][[2]]}, {i, 1, endtime}],
  PlotStyle → {RGBColor[0.4, 0.8, 0.4]}],
 ListLinePlot[{tabBA1, tabBA11, tabBA2, tabBA45, tabBA5, tabBQ, tabXBB15, tabXBBO},
  PlotStyle → {colourBA1, colourBA11, colourBA2, colourBA45,
    colourBA45, colourBQ, colourXBB15, colourXBBO}, PlotRange → All,
      Filling → {1 → {2}, 2 → {3}, 3 → {4}, 4 → Axis,
    5 → {6}, 6 → {7}, 7 → {8}, 8 → Axis}],
 (*Graphics[Text["AB",{5,95},{-1,0}]],*)
 ListPlot[{{starttime, 0}, {starttime, 100}},
  Joined → True, PlotStyle → {Black, Thickness[0.002], Dashed}],
 PlotRange → {{1, curdate["2023-06-01"]}, {0, graphheight}},
 Ticks → {xaxis, Automatic},
 AspectRatio → 0.6,
 ImageSize → 450
]
```

Out[◦]=


100
80
60
40
20
2022
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
2023
Feb
Mar
Apr
May
June