732A76_Research_Project_report

FHM's daily COVID19 reports

Chenjian Shi

2022/01/05

Abstract

In this paper, we could introduce download FHM's covid19 data excel and the process of these data and analysis of the outputs.

Contents

1 Introduction

The first part of the project is to download (appropriately naming) Folkhalsomyndigheten's daily (for those days that it was provided) excel spreadsheet report concerning COVID19 in Sweden from Wayback machine. Then, to design a database (in R) that stores all the the information in those sheets, implement a program (R) that reads in all of the sheets. The program should read these sheets from a given directory on the hard drive. The program should be flexible in the sense to read all the files in the given directory, not have the file names hardcoded, as new files can be added at a later stage. The database and program have to be flexible as sheet names in the excel files evolve, e.g. new ones are added can be renamed. They should be flexible for the future if FHM adds new, renames, removes sheets. The R database should be designed in such a way that is easy to use, with unproblematic extraction of the data from it. As part of the final report the downloaded spreadsheets and all the R code need to be handed-in as publicly available data and R package on GitHub or CRAN.

2 Data Access

The data is downloaded from Folkhalsomyndigheten's daily excel spreadsheet report concerning COVID19 in Sweden from Wayback machine. In order to access the data, here we used a python script which implemented by a package called selenium(Conservancy n.d.). The selenium we used in this project is the Selenium WebDriver. This is a kind of browser-based automation test suite, which allows python scripts to run the automation tests like mouse-clicking, etc. by locating the XPath elements of the website, etc.

In this paper, we would use the chrome driver(version 96.0.4664.45, based on Chrome 96.0.4664.110) which download from Chrome Driver's official website(Chromium and teams n.d.), which is the webdriver for Chrome.

2.1 Config

Before running the main script, we first need to do some settings in the config.py file.

```
# Config.py

# Location of chrome driver
chrome_driver = r"C:\Program Files (x86)\Google\Chrome\Application\
                  chromedriver.exe"

# Url for Wayback Machine
Wayback_Machine_URL = 'https://web.archive.org/'

# Target URL
Target_URL = ('https://www.folkhalsomyndigheten.se/smittskydd-beredskap/utbrott
              /aktuella-utbrott/covid-19/'
              'bekraftade-fall-i-sverige',
              'https://www.folkhalsomyndigheten.se/smittskydd-beredskap/utbrott/
              aktuella-utbrott/covid-19/' \
             'statistik-och-analyser/bekraftade-fall-i-sverige/')

# Define the start date and final date
start_date = 'APR 04 2020'
final_date = 'JAN 19 2022'

# Download directory
Download_directory = r'G:\LiU\732A76_Research_Project\data'
```

In this file, we need to set the location of the Chrome webdriver first, which helps python scripts to run the

webdriver. Besides, we set the URL for the Wayback Machine and the URL for the Folkhalsomyndigheten in order to get the files from the past. Then, we set the start and end dates of the files we need, which are in form of 'MON DAY YEAR'. At last, we set the download directory for the files.

2.2 Data file downloads

After finishing the setting of configs.py, we just run the 'main.py' file so that the download would start. The process of the downloads(in PyCharm IDE) would be like this

```
Jan 02 2022 - 23:25:14, Start to downloading files
Jan 02 2022 - 23:26:18, Finished downloading of the file in Sep 17 2021
Jan 02 2022 - 23:27:20, Finished downloading of the file in Sep 18 2021
```

If there is no data at that date or failed the downloads, the script would shows like this

```
Jan 02 2022 - 23:25:14, Start to downloading files
Jan 02 2022 - 23:26:18, failed to download the file in Sep 17 2021
Jan 02 2022 - 23:27:20, No file could be download in Sep 18 2021
```

Each download may take about 1 minute, which depends on the network situation. The total time used in downloading the files from APR 03 2020 to DEC 28 2021 takes about 10 hours to download.

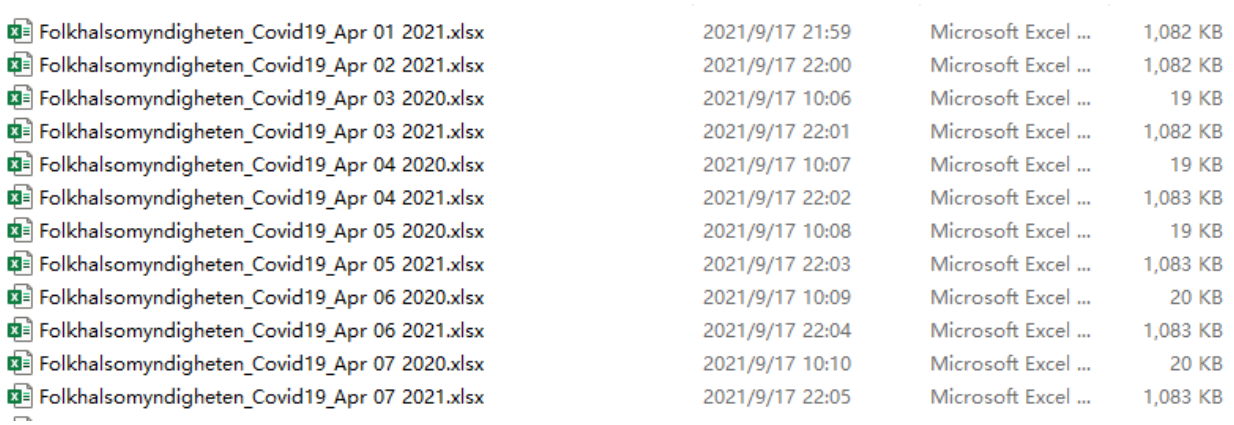


Figure 1: Data files(parts)

3 Data Proccessing

After we get the data files like above, we start to do the process these data by R script.

The required packages for the script are below:

```
library(readxl)
library(tidyverse)
library(plotly)
library(config)
library(htmlwidgets)
library(processx)
```

3.1 Config

Before starting the process of data we need to do the config.yml.

```
default:
  start_time: "2020/04/03"
  end_time: "2022/01/19"
  file_add_and_name: './data/Folkhalsomyndigheten_Covid19_'
  data_update_seq: 14
  file_type: '.xlsx'
  fig_output_type: '.png'
  resolution: !!seq [1920,1080]
  # Sheet type
  Antal per dag region: !!seq [1,1,2,23]
  Antal avlidna per dag: !!seq [1,1,2,2]
  Antal intensivvårdade per dag: !!seq [1,1,2,2]
  Totalt antal per region: !!seq [2,1,2,5]
  Totalt antal per kön: !!seq [2,1,2,4]
  Totalt antal per åldersgrupp: !!seq [2,1,2,4]
  Veckodata Region: !!seq [3,3,4,11]
  Veckodata Kommun_stadsdel: !!seq [3,6,7,10]
  Veckodata Riket: !!seq [3,0,3,13]
  # Combined sheet
  Antal per dag region_on_kön: !!seq [4,0,4,5]
  Antal per dag Åldersgrupp_on_kön: !!seq [4,0,6,5]
  Antal per dag region_on_Åldersgrupp: !!seq [4,0,4,6]
```

We first set the start/end date of the file we download, then set the file address and the name, then of course the file type. In this case, the file address with time sequence would create automatically like this "./data/Folkhalsomyndigheten_Covid19_Apr 03 2021.xlsx" later.

About sheet type, here we have defined 4 types of sheets according to the current data. For now, I would first explain what these sequences doing. The first digit stands for the type of the sheets, the second letter stands for the column of time/label depending on the type of sheet. The third letter stands for the start column of the data, and the fourth letter stands for the end of the data.

For example, Antal per dag region: !!seq [1,1,2,23], means it is a type 1 sheet, and the time column is in the first column, the data we need starts from the second column to the twenty-third column.

Then there is a config named "data_update_seq", this is about the correction date of the data from the first seen to the fixed data. The default date 14 is defined for most cases, the most 14 days data usually changes.

Besides, there is a config about "fig_output_type" and "resolution", these configs are used for image output of the HTML file, which is actually like taking a screenshot of the .html file.

And this is the way offered by the plotly official(https://plotly.com/r/static-image-export/).

In this case, the user needs to install the orca by the following instruction: https://github.com/plotly/orca#installation.

And, the resolution of the output picture is about 1920x1080, but the user could change it as they want, but the user should know that, if the resolution is too low, the legend may be not showing completely. If the user meets this situation, please try to improve the resolution to like 4,096 x 2,160, etc.

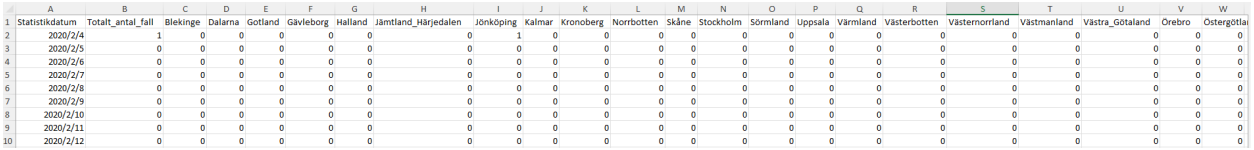
By the way, if users want to get the result with specific data, I suggest it changes the config and runs the program.

3.2 Data reading

3.2.1 Type 1 sheet

According to the settings: Antal per dag region: !!seq [1,1,2,23]

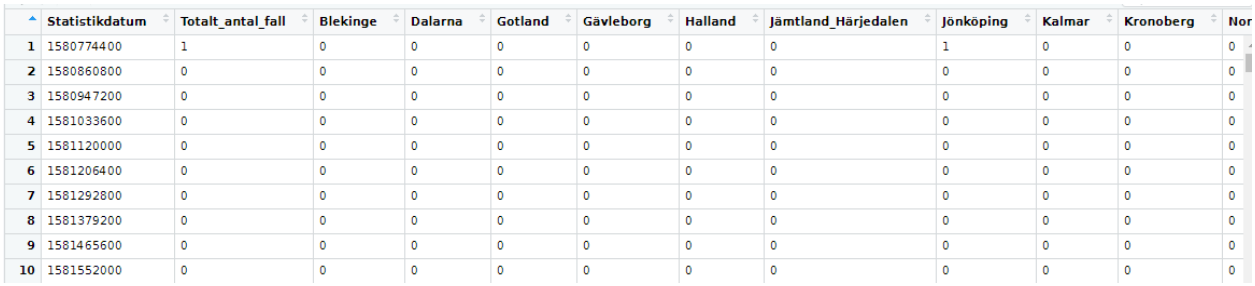
The sheet (Antal per dag region) is a kind of type 1 sheet, with time sequence columns and data columns.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Statistikdatum	Totalt_antal_fall	Blekinge	Dalarna	Gotland	Gävleborg	Halland	Jämtland_Härjedalen	Jönköping	Kalmar	Kronoberg	Norrbotten	Skåne	Stockholm	Sörmland	Uppsala	Värmland	Västerbotten	Västernorrland	Västmanland	Västra_Götaland	Örebro	Östergötla
2	2020/2/4	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2020/2/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2020/2/6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2020/2/7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2020/2/8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	2020/2/9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	2020/2/10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	2020/2/11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	2020/2/12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2: Type 1 sheet(Antal_per_dag_region)

After relative processing, we could get the data frame with the 1st column as a time column, and the rest 2nd to 23rd column as data.



	Statistikdatum	Totalt_antal_fall	Blekinge	Dalarna	Gotland	Gävleborg	Halland	Jämtland_Härjedalen	Jönköping	Kalmar	Kronoberg	Nor
1	1580774400	1	0	0	0	0	0	0	1	0	0	0
2	1580860800	0	0	0	0	0	0	0	0	0	0	0
3	1580947200	0	0	0	0	0	0	0	0	0	0	0
4	1581033600	0	0	0	0	0	0	0	0	0	0	0
5	1581120000	0	0	0	0	0	0	0	0	0	0	0
6	1581206400	0	0	0	0	0	0	0	0	0	0	0
7	1581292800	0	0	0	0	0	0	0	0	0	0	0
8	1581379200	0	0	0	0	0	0	0	0	0	0	0
9	1581465600	0	0	0	0	0	0	0	0	0	0	0
10	1581552000	0	0	0	0	0	0	0	0	0	0	0

Figure 3: Type 1 data(Antal_per_dag_region)

3.2.2 Type 2 sheet

According to the settings, Totalt antal per kön: !!seq [2,1,2,4]

the sheet (Totalt antal per kön) is a kind of type 2 sheet, with only data.

1	Region	Totalt_antal_fall	Fall_per_100000_inv	Totalt_antal_intensivvårdade	Totalt_antal_avlidna
2	Blekinge	14158	8890	65	131
3	Dalarna	29518	10257	194	359
4	Gotland	4754	7934	34	54
5	Gävleborg	38964	13555	252	566
6	Halland	44945	13380	178	329
7	Jämtland Härjedalen	13515	10314	71	128
8	Jönköping	46645	12798	347	567
9	Kalmar	26950	10961	105	255
10	Kronoberg	24076	11921	126	328
11	Norrbotten	28826	11542	224	295

Figure 4: Type 2 sheet(Totalt_antal_per_region)

In this case, the second letter does not stand for a time, but the label. Then we just add one column(Date of the file) to the data frame as below, then combine these data into one named final_Totalt_antal_per_region.

	Date	Region	Totalt_antal_fall	Fall_per_100000_inv	Totalt_antal_intensivvårdade	Totalt_antal_avlidna
1	2020/04/03	Blekinge	27	16.9166564941406	2	1
2	2020/04/03	Dalarna	147	51.0476913452148	10	14
3	2020/04/03	Gotland	12	20.1052169799805	1	0
4	2020/04/03	Gävleborg	147	51.1514282226562	12	4
5	2020/04/03	Halland	114	34.1472778320312	10	8
6	2020/04/03	Jämtland Härjedalen	84	64.2152709960938	1	0
7	2020/04/03	Jönköping	187	51.4302825927734	15	13
8	2020/04/03	Kalmar	46	18.7413940429688	2	1
9	2020/04/03	Kronoberg	45	22.3359432220459	NA	0
10	2020/04/03	Norrbotten	73	29.1891422271729	10	1

Figure 5: Type 2 data(Totalt_antal_per_region)

3.2.3 Type 3 sheet

For type 3, there are two types of data, but both of them are the same for the time part with year column and week column

First, the sheet like Veckodata Region contains the week data and multiple labels(like KnKod, etc.)

	A	B	C	D	E	F	G	H	I	J
1	år	veckonummer	KnKod	KnNamn	Stadsdel	Kommun_stadsdel	tot_antal_fall_per10000inv	antal_fall_per10000_inv	tot_antal_fall	nya_fall_vecka
2	2020	1	1440	Ale		Ale	0	0	0	0
3	2020	2	1440	Ale		Ale	0	0	0	0
4	2020	3	1440	Ale		Ale	0	0	0	0
5	2020	4	1440	Ale		Ale	0	0	0	0
6	2020	5	1440	Ale		Ale	0	0	0	0
7	2020	6	1440	Ale		Ale	0	0	0	0
8	2020	7	1440	Ale		Ale	0	0	0	0
9	2020	8	1440	Ale		Ale	0	0	0	0
10	2020	9	1440	Ale		Ale	0	0	0	0

Figure 6: Type 3-1 sheet(Veckodata Kommun_stadsdel)

So according to the setting Veckodata Kommun_stadsdel: !!seq [3,6,7,10], choose the chosen label column and data columns as below.

	år	veckonummer	Kommun_stadsdel	tot_antal_fall_per10000inv	antal_fall_per10000_inv	tot_antal_fall	nya_fall_vecka
1	2020	1	Ale	0	0	0	0
2	2020	2	Ale	0	0	0	0
3	2020	3	Ale	0	0	0	0
4	2020	4	Ale	0	0	0	0
5	2020	5	Ale	0	0	0	0
6	2020	6	Ale	0	0	0	0
7	2020	7	Ale	0	0	0	0
8	2020	8	Ale	0	0	0	0
9	2020	9	Ale	0	0	0	0
10	2020	10	Ale	0	0	0	0

Figure 7: Type 3-1 data(Veckodata Kommun_stadsdel)

Then for the sheet like Veckodata_Riket, it only contains the time(year and week) column and data columns, in this case, we set the second letter as 0.

	A	B	C	D	E	F	G	H	I	J	K	L
1	år	veckonummer	Antal_fall_vecka	Antal_fall_100000inv_vecka	Antal_fall_100000inv_14dagar	Kum_antal_fall	Kum_fall_100000inv	Antal_nyaintensivvårdade_vecka	Kum_antal_intensivvårdade	Antal_avlidna_vecka	Antal_avlidna_milj_inv_vecka	Kum_antal_avlidna
2	2020	6	1	0	0	1	0	0	0	0	0	0
3	2020	7	0	0	0	1	0	0	0	0	0	0
4	2020	8	0	0	0	1	0	0	0	0	0	0
5	2020	9	13	0	0	14	0	0	0	0	0	0
6	2020	10	211	2	2	225	2	3	3	0	0	0
7	2020	11	838	8	10	1063	10	16	19	5	1	5
8	2020	12	924	9	17	1987	19	88	107	44	4	49
9	2020	13	1957	19	28	3944	38	238	345	190	18	239
10	2020	14	3229	31	50	7173	69	277	622	451	44	690

Figure 8: Type 3-2 sheet(Veckodata_Riket)

So according to the settings, Veckodata Riket: !!seq [3,0,3,13], we could get the data like below

	år	veckonummer	Antal_fall_vecka	Antal_fall_100000inv_vecka	Antal_fall_100000inv_14dagar	Kum_antal_fall	Kum_fall_100000inv	Antal_nyaintensivvår
1	2020	6	1	0	0	1	0	0
2	2020	7	0	0	0	1	0	0
3	2020	8	0	0	0	1	0	0
4	2020	9	13	0	0	14	0	0
5	2020	10	211	2	2	225	2	3
6	2020	11	838	8	10	1063	10	16
7	2020	12	924	9	17	1987	19	88
8	2020	13	1957	19	28	3944	38	238
9	2020	14	3229	31	50	7173	69	277
10	2020	15	3740	36	67	10913	106	289

Figure 9: Type 3-2 data(Veckodata_Riket)

3.2.4 Type 4 sheet

For the type 4 sheet, it means the combined sheet, which takes the main sheet as base then using the data from other sheet(vice sheet) as a reference to create new data groups, for example, like Antal per dag Åldersgrupp_on_kön: !!seq [4,0,6,5]

In this sheet(Antal per dag Åldersgrupp_on_kön), we use sixth sheet(Totalt antal per åldersgrupp) as the main sheet, and use fifth sheet (Totalt antal per kön) as vice sheet. In this case, the main data is based on the Totalt antal per åldersgrupp

	Åldersgrupp	Totalt_antal_fall	Totalt_antal_intensivvårdade	Totalt_antal_avlidna
2	Ålder_0_9	39635	37	9
3	Ålder_10_19	172498	55	5
4	Ålder_20_29	207382	216	26
5	Ålder_30_39	212857	377	48
6	Ålder_40_49	217990	877	124
7	Ålder_50_59	185815	1728	392
8	Ålder_60_69	94303	2275	1078
9	Ålder_70_79	43946	1989	3404
10	Ålder_80_89	26038	449	6159
11	Ålder_90_plus	11521	10	3924
12	Uppgift saknas	160	1	1

Figure 10: Type 4 main sheet(Totalt antal per åldersgrupp)

	A	B	C	D
1	Kön	Totalt_antal_fall	Totalt_antal_intensivvårdade	Totalt_antal_avlidna
2	Man	595297	5604	8299
3	Kvinna	615700	2404	6860
4	Uppgift saknas	1148	6	11

Figure 11: Type 4 main sheet(Totalt antal per kön)

Then combined the data with the vice sheet(Totalt antal per kön), which provides the rates of different sex takes, which means we could get the data of Totalt_antal_fall, Totalt_antal_intensivvårdade and Totalt_antal_avlidna with different sex and age group.

	Date	Åldersgrupp	Totalt_antal_fall_Man	Totalt_antal_fall_Kvinna	Totalt_antal_fall_Uppgift saknas	Totalt_antal_intensivvårdade_Man	Totalt_antal_intensivvårdad
1	2020/04/03	Ålder_0_9	17.7879236590984	17.2063178677196	0.00575847318196775	NA	NA
2	2020/04/03	Ålder_10_19	68.6105626850938	66.3672260612043	0.0222112537018756	NA	NA
3	2020/04/03	Ålder_20_29	220.57025337282	213.358341559724	0.0714050674564001	NA	NA
4	2020/04/03	Ålder_30_39	278.508061862455	269.401776900296	0.0901612372490951	NA	NA
5	2020/04/03	Ålder_40_49	407.597564988483	394.270483711747	0.131951299769661	NA	NA
6	2020/04/03	Ålder_50_59	570.230009871668	551.585389930898	0.184600197433366	NA	NA
7	2020/04/03	Ålder_60_69	443.17341230668	428.683119447187	0.143468246133597	NA	NA
8	2020/04/03	Ålder_70_79	446.22277064824	431.632773938796	0.144455412964791	NA	NA
9	2020/04/03	Ålder_80_90	442.15695952616	427.699901283317	0.143139190523198	NA	NA
10	2020/04/03	Ålder_90_plus	192.617801908523	186.319842053307	0.0623560381704508	NA	NA
11	2020/04/03	Uppgift saknas	1.52467917077986	1.47482724580454	0.000493583415597236	NA	NA

Figure 12: Type 4 data(Antal_per_dag_Åldersgrupp_on_kön)

3.3 Output

After finishing processing the data, we would use the package plotly to do the visualization. Then save the plots in form of HTML, for example Antal_per_dag_region. Then a screen shot of the html file would be taken, then save the format as defined above(default png).

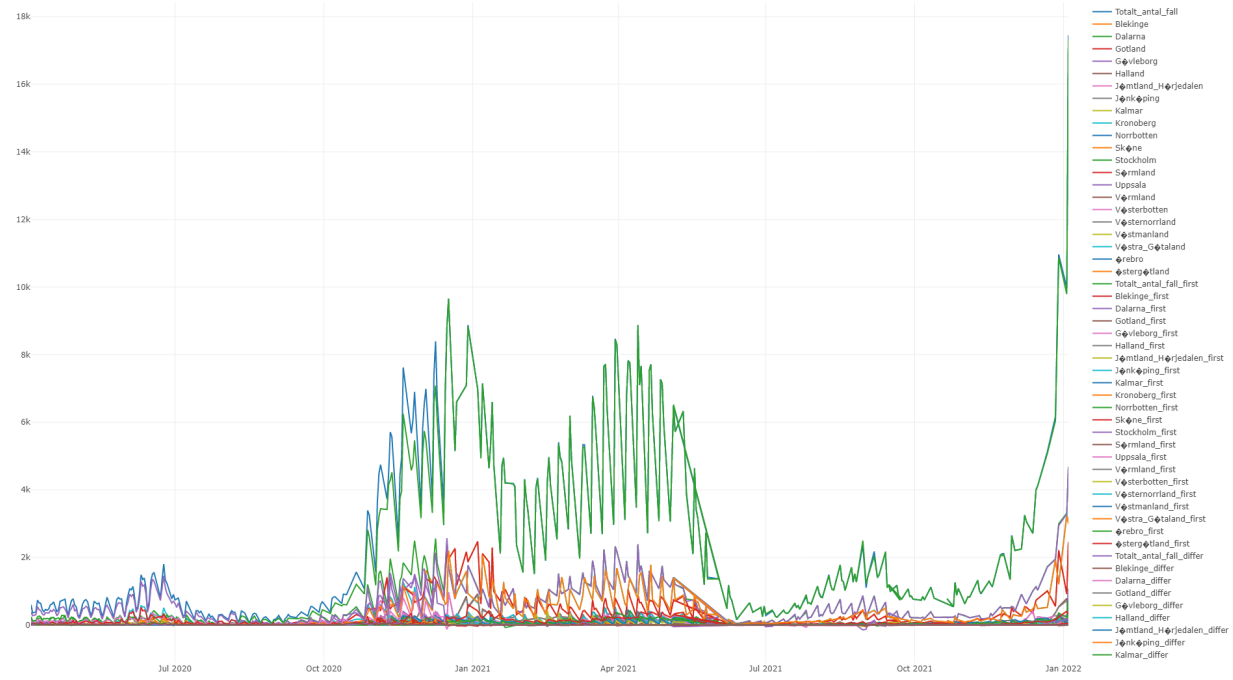


Figure 13: Antal_per_dag_region

4 Result Anlysis

The plots below contain different labels, for labels, which contains '_first' means the data first recorded in the file, and the '_differ' is the difference between the first seen date data and the data after the correction cycle.

4.1 Antal_avlidna_per_dag

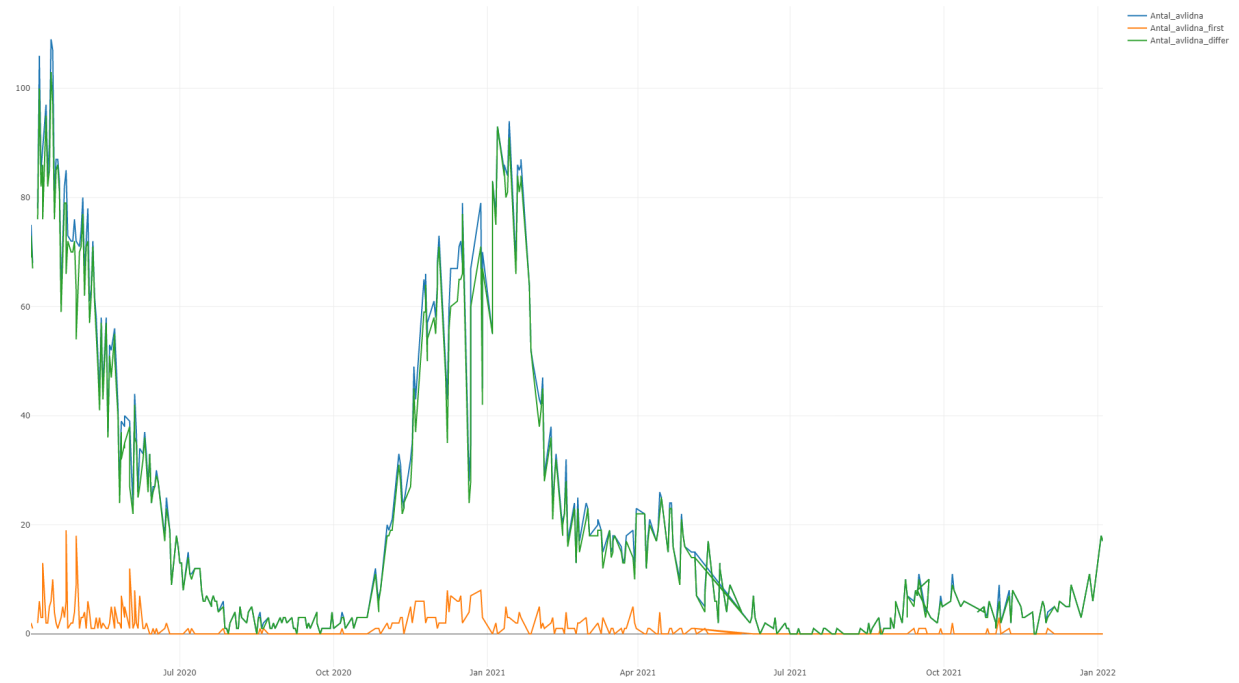


Figure 14: Antal_avlidna_per_dag

According to the plot, we could find that the number of the death per day presents as a seasonal like data, which takes two peaks in Apr 2020 and Jan 2021. But the third peaks does appear in Oct 2021. Besides, the number of death per day is kind of relative to the weather, for example, it gets lower value from Apr to Oct, which is summer time and gets higher value from Oct to Jan when the temperature gets lower.

4.2 Antal_per_dag_region

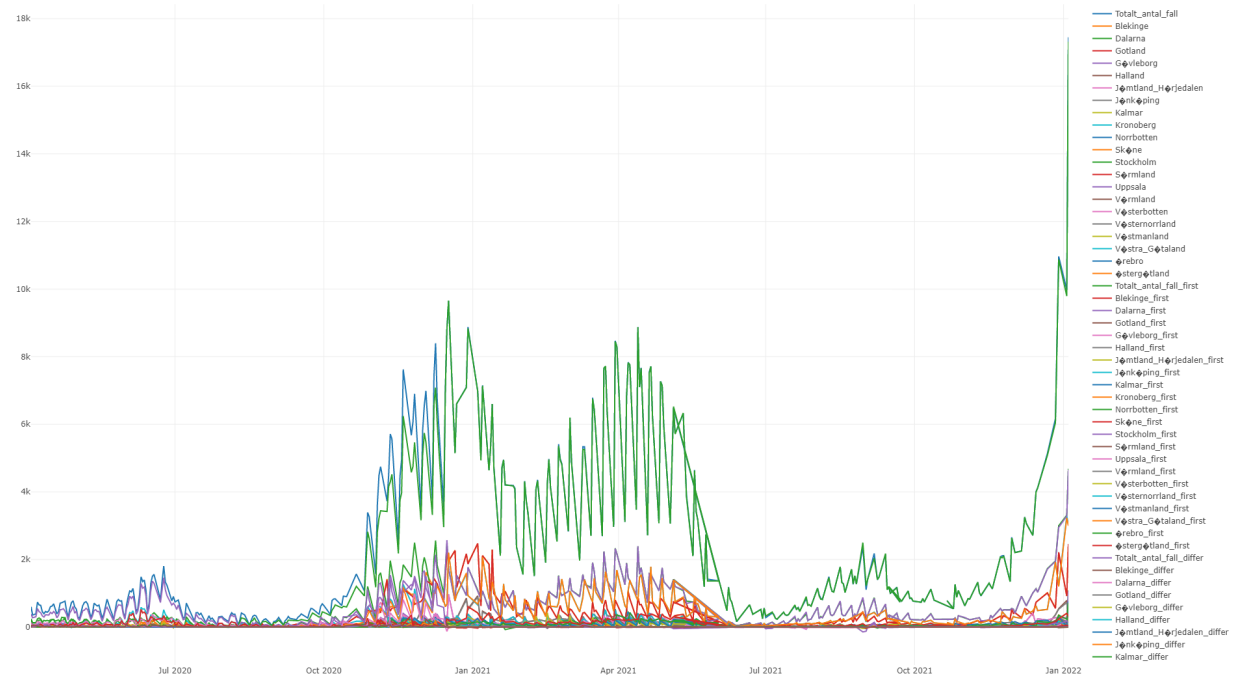


Figure 15: Antal_per_dag_region

According to the plot, we could find that the number of cases per day presents a seasonal like data, which from Oct 2020 to Oct 2021. The peeks occurs from Oct 2020 to Apr 2021, but there is a trough in the middle from Jan 2021 to Mar 2021.

By filtering values and comparing, we could find that the most cases takes place in Skane, Stockholm and Vastra Gotaland.

4.3 Antal_intensivvårdade_per_dag

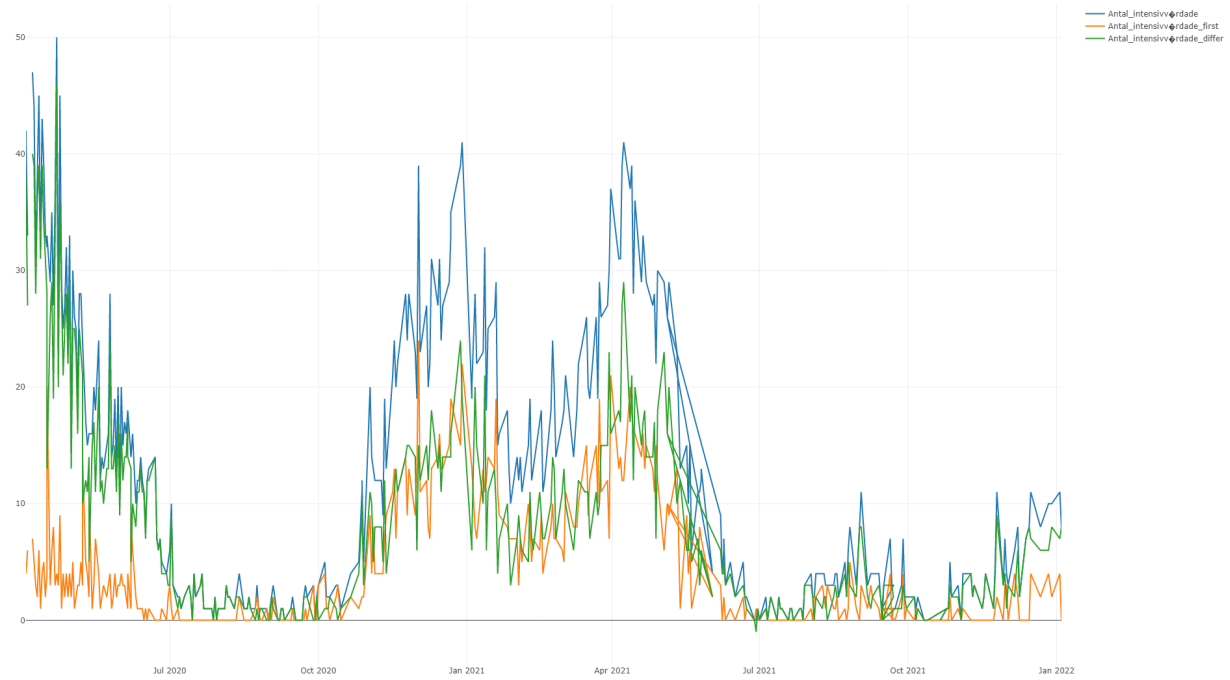


Figure 16: Antal_intensivvårdade_per_dag

According to the plot, we could find that the number of intensive care per day presents a data without any seasonal, which the peaks occurs in May 2020, Dec 2021 and Apr 2021. This data is similar to the data of Antal_avlidna_per_dag from Apr 2020 to Mar 2021, but from Apr 2021 it turns to a different trend by comparing with the Antal_avlidna_per_dag. This may caused by the Antal_per_dag_region peaks at Apr 2021.

4.4 Totalt_antal_per_åldersgrupp

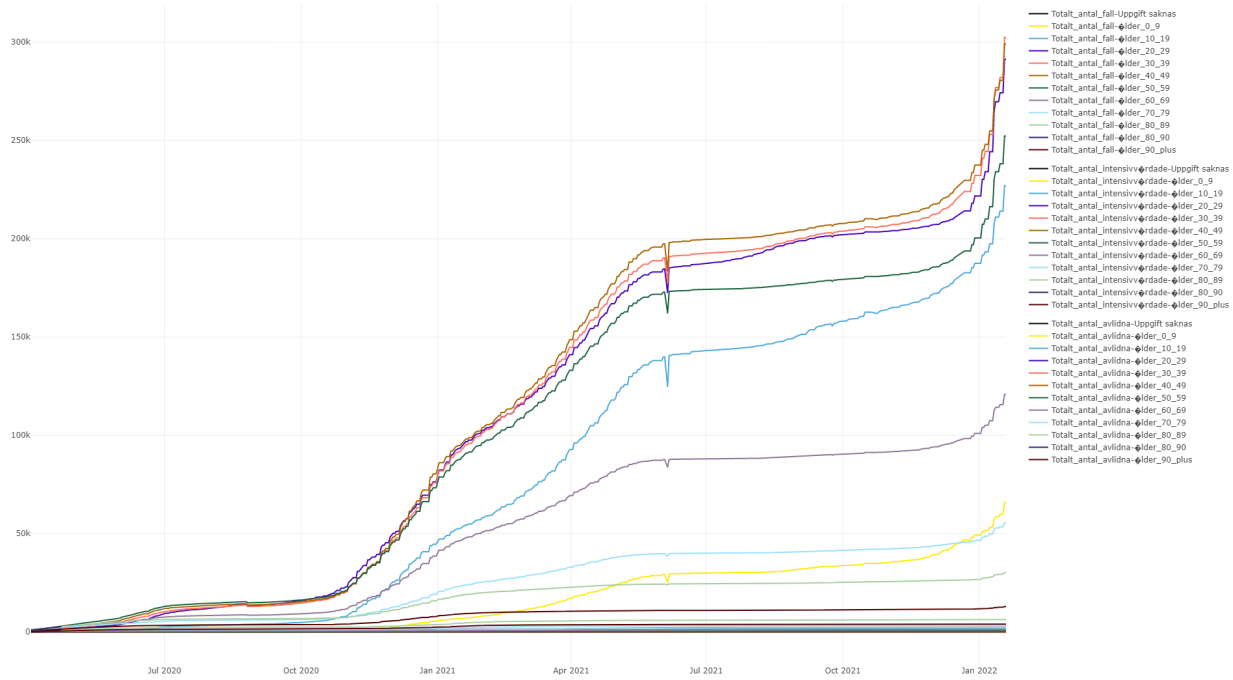


Figure 17: Totalt_antal_per_åldersgrupp

Here is the plot that contains the all the categories.

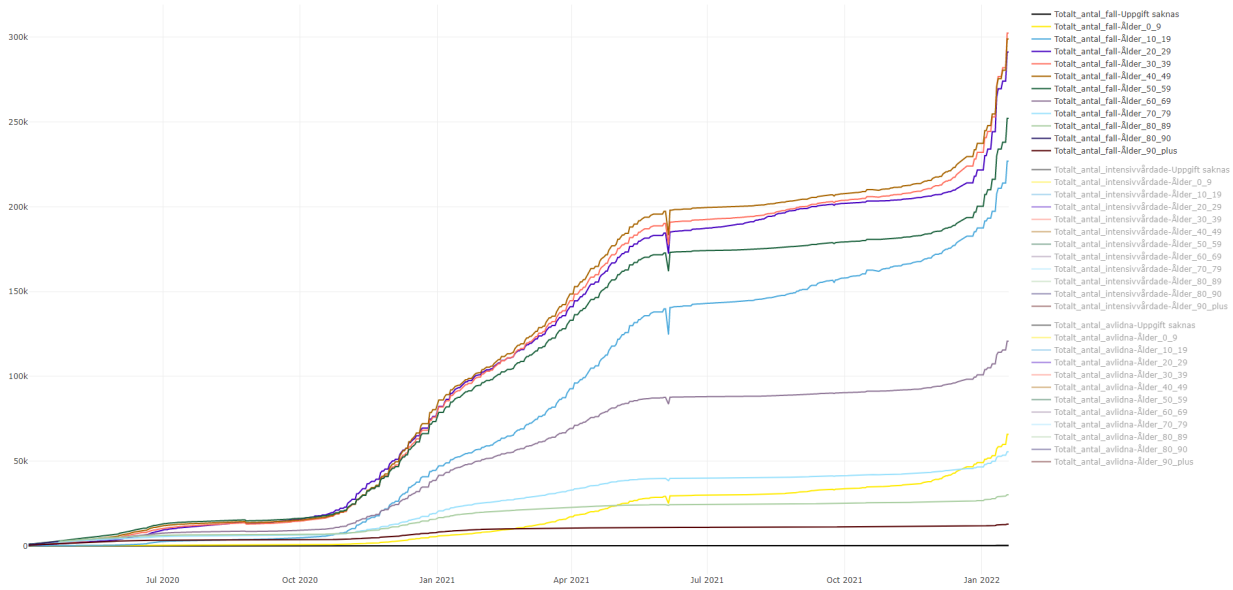


Figure 18: Totalt_antal_per_åldersgrupp-fall

According to the plot, we could find that for all the age group, the trend of total number of cases is growing up. Besides, the people in age group 20-49 takes most cases, and the age group 10-19 and age group 50-59 take the middle value and the age group 70-90 plus and 0-9 stays in a low value for a long time.

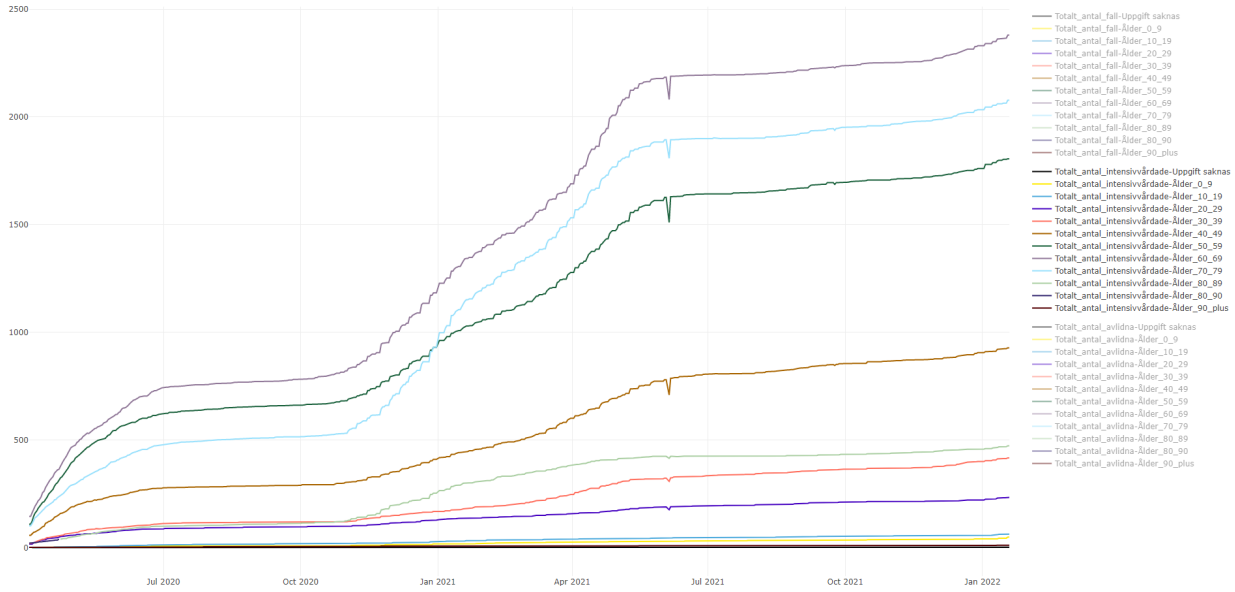


Figure 19: Totalt_antal_per_åldersgrupp-ICU

As for number of intensive care per day, most people are in age from 50-80 (age group 60-69 top then 70-79, 50-59 at third).

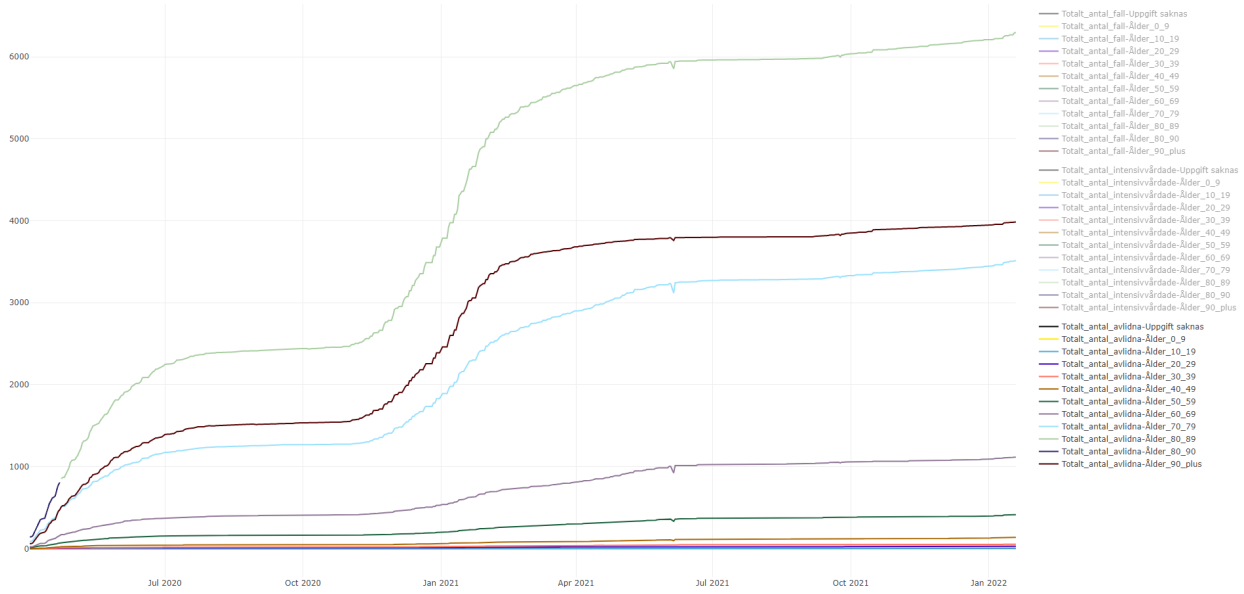


Figure 20: Totalt_antal_per_åldersgrupp-avlidna

As for number of death per day, most people are in age from 80-89 then age group 90 plus and 70-79 takes the middle level.

4.5 Totalt_antal_per_kön

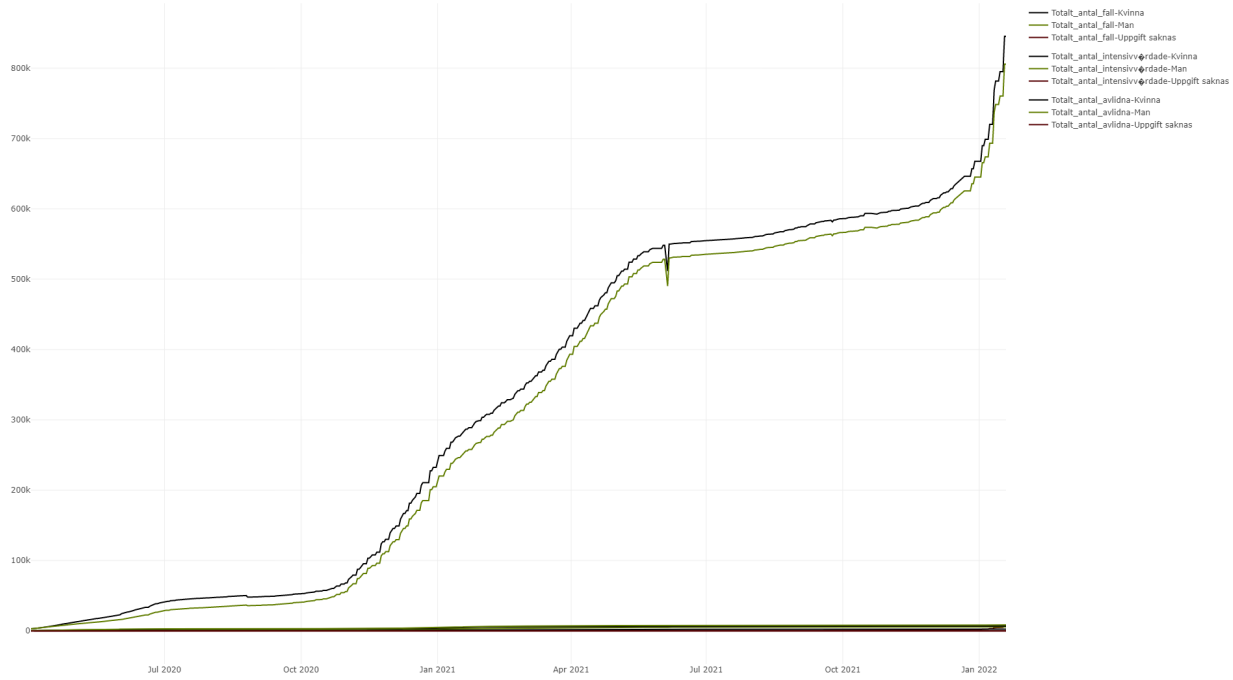


Figure 21: Totalt_antal_per_kön

According to the plot, we could find that the woman take more risk of getting effected than the man. But the death rate and the intensive care rate the man is lower the woman.

4.6 Totalt_antal_per_region

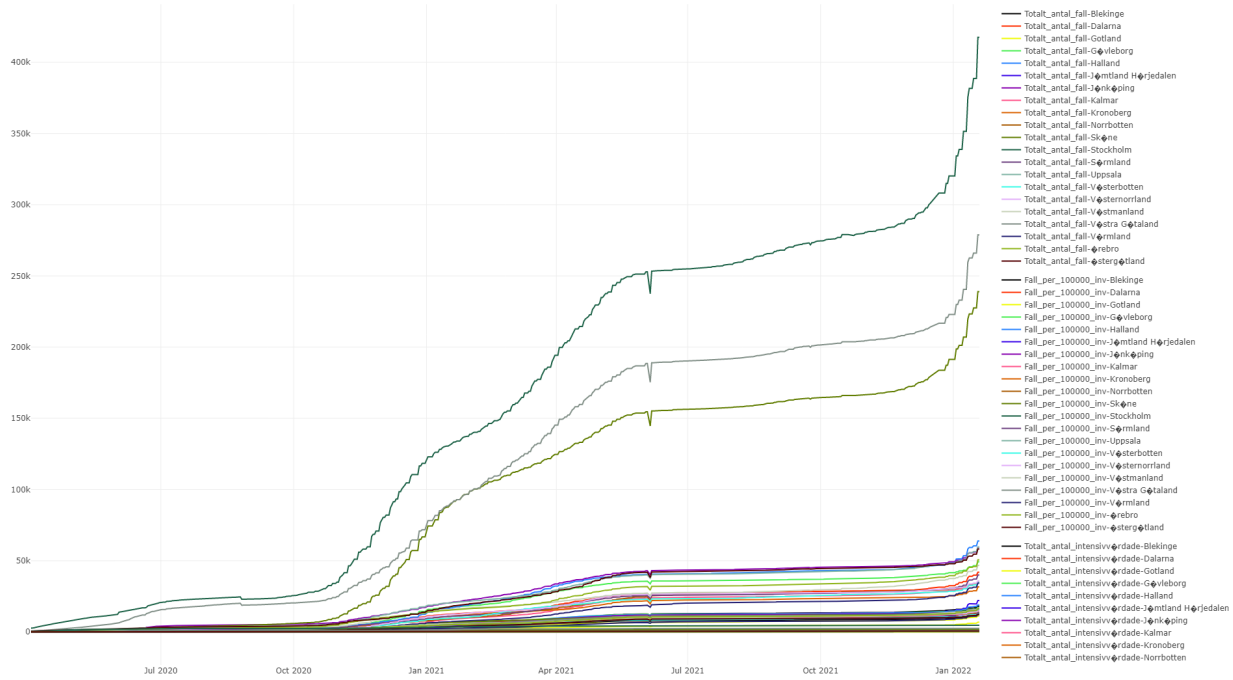


Figure 22: Totalt_antal_per_region

According to the plot, we could find that the trend of the data is similar to the data we mentioned previously, for example, like number of cases, most cases takes place in Skane, Stockholm and Vastra Gotaland,etc.

But for per 100000 infected, the rate of most regian are the same.

As for total number of intensive care, it has the same rate as number of cases does, which means most cases takes place in Skane, Stockholm and Vastra Gotaland.

As for total number of death, it is also Skane, Stockholm and Vastra Gotaland take the top 3 rates.

4.7 Veckodata_Kommun_stadsdel

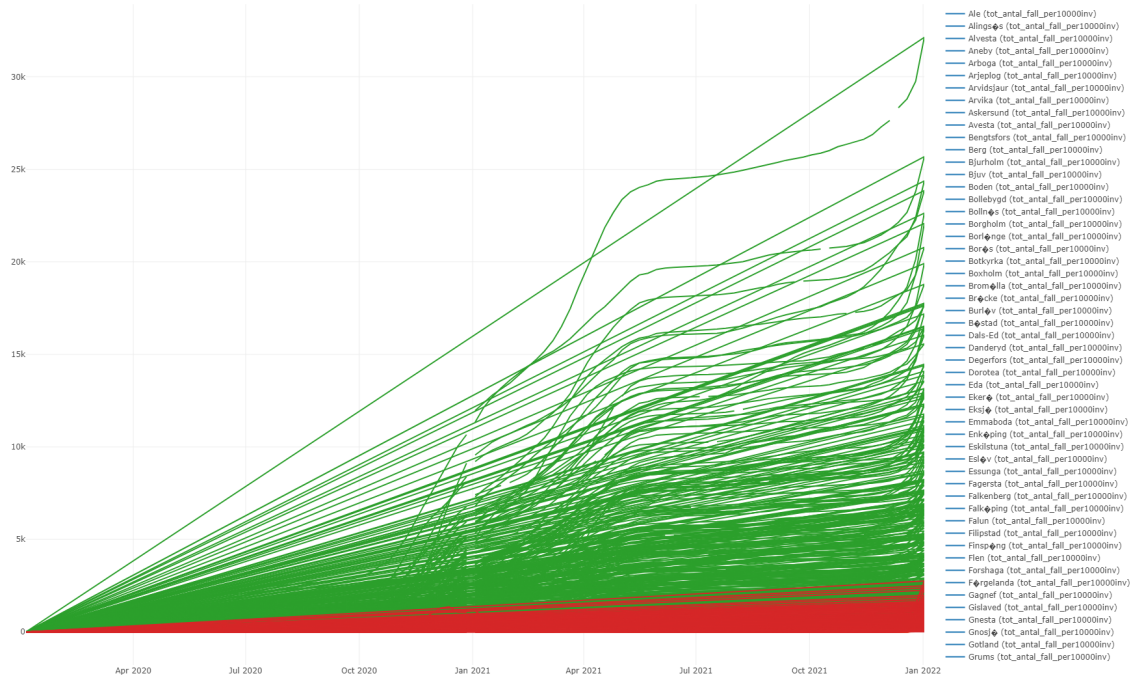


Figure 23: Veckodata_Kommun_stadsdel

It is hard to find anything in this plot.

4.8 Veckodata_Region

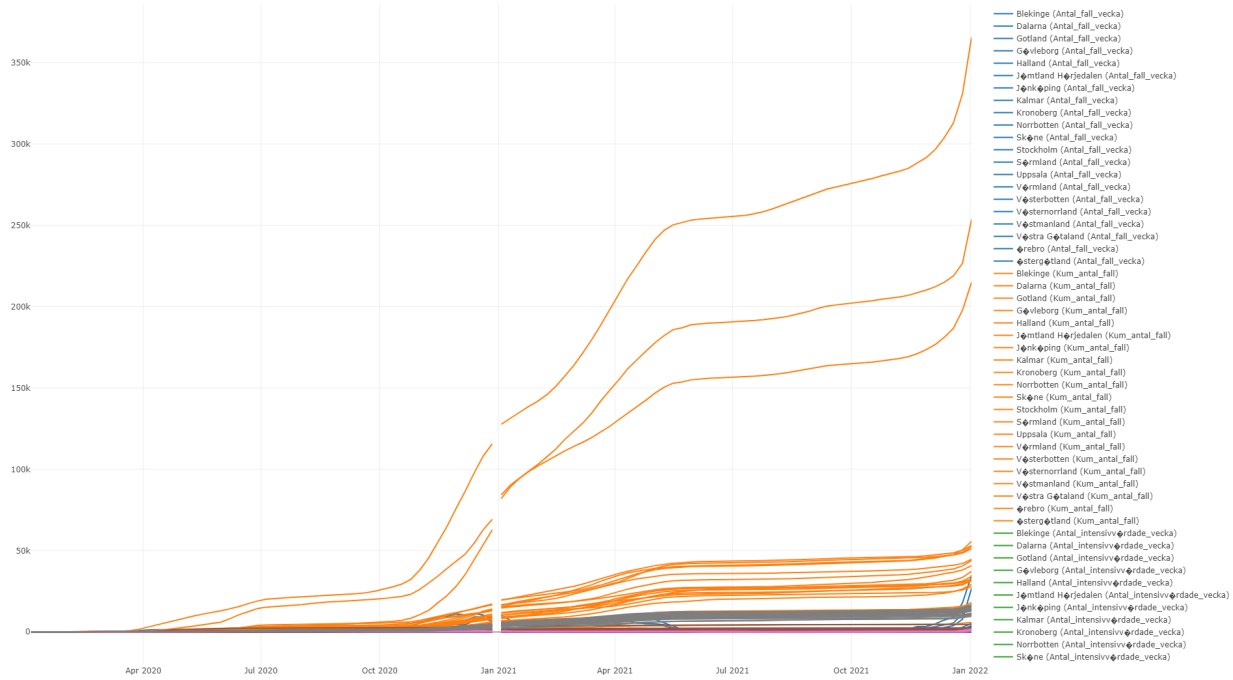


Figure 24: Veckodata_Region

This plot is similar to the Antal_per_dag_region.

4.9 Veckodata_Riket

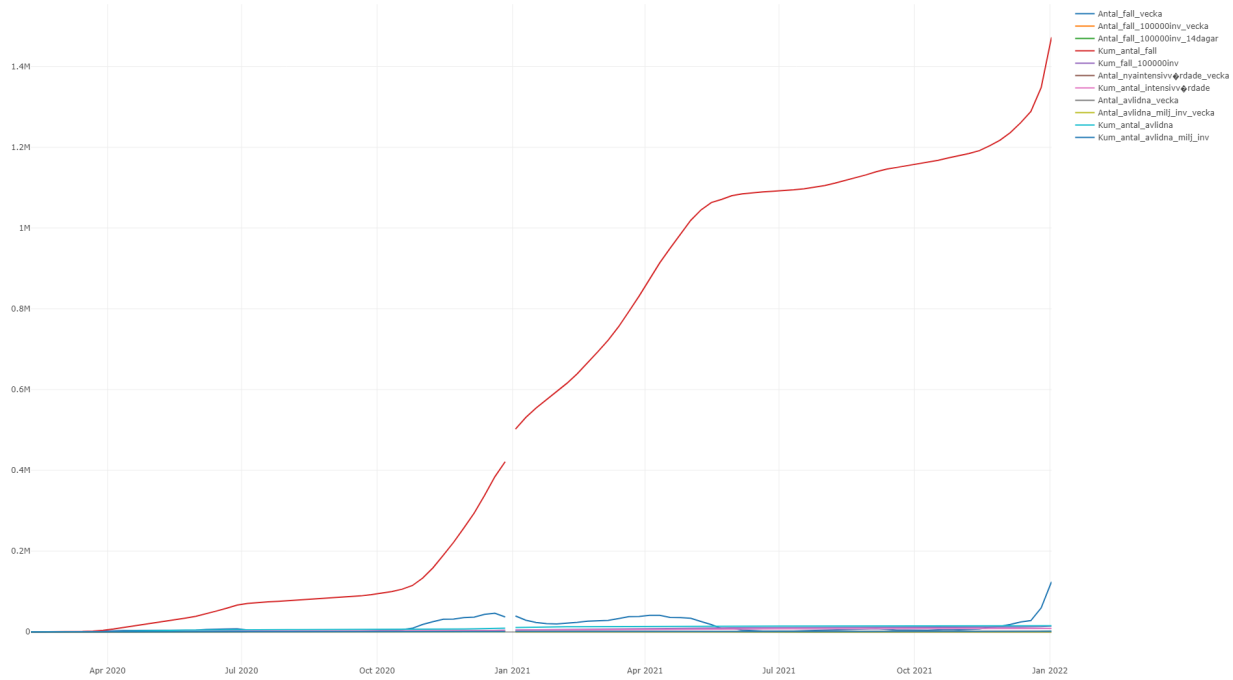


Figure 25: Veckodata_Riket

The trend of data in this plot are same to the other plots.

4.10 Antal_per_dag_Åldersgrupp_on_kön

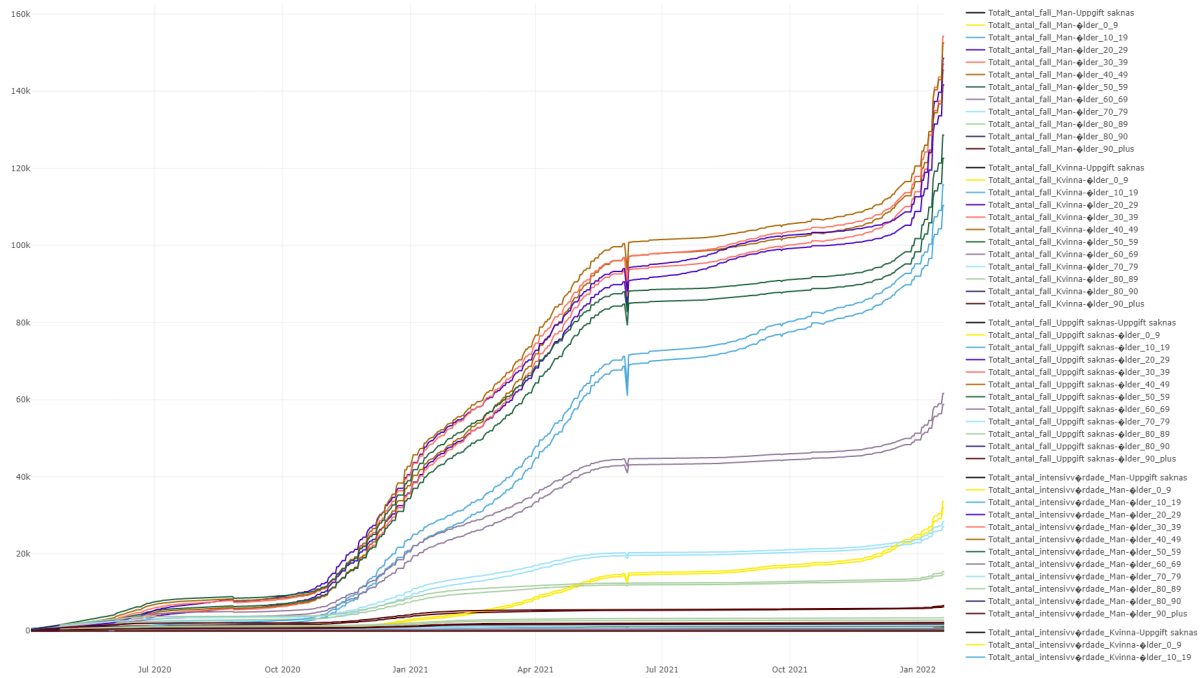


Figure 26: Antal_per_dag_Åldersgrupp_on_kön

Based on the result of Totalt_antal_per_åldersgrupp, we could find that for each age group, the woman take the most part.

4.11 Antal_per_dag_region_on_Åldersgrupp

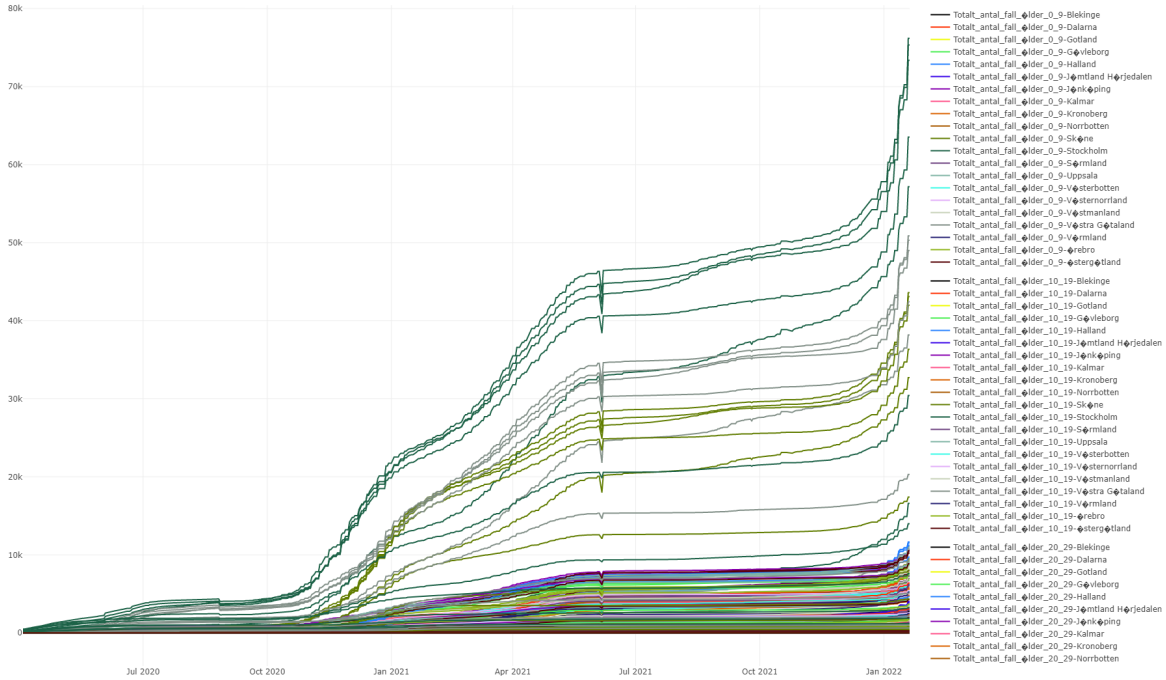


Figure 27: Antal_per_dag_region_on_Åldersgrupp

Based on the result of Totalt_antal_per_region, we could find that for each region, the age group 40-49 take the most part.

4.12 Antal_per_dag_region_on_kön.html

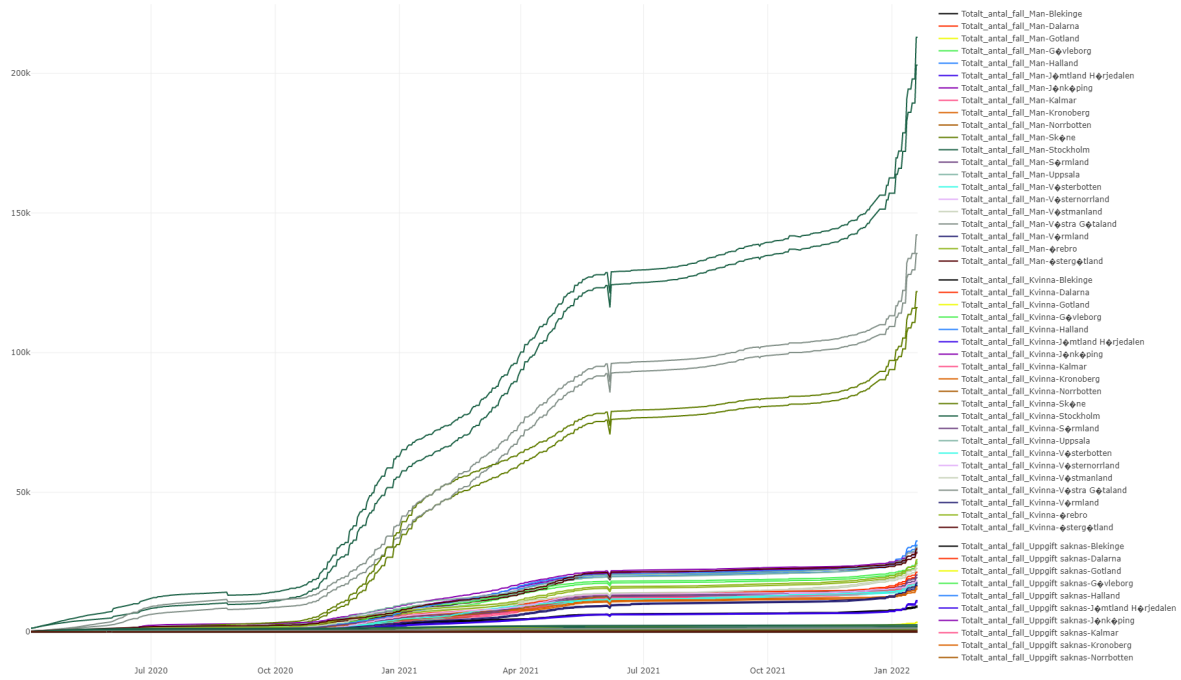


Figure 28: Antal_per_dag_region_on_kön

Based on the result of Totalt_antal_per_region, we could find that for each region, the woman take the most part.

5 Appendix

All code and files is uploaded in the Github.

Data access

https://github.com/ChenjianS47/WayBackMachine_FHM_COVID-19_statistics

Data process

https://github.com/ChenjianS47/FHMs_daily_COVID-19_reports

References

Chromium, and WebDriver teams. n.d. “ChromeDriver - WebDriver for Chrome.” Accessed December 30, 2021. https://chromedriver.chromium.org/.

Conservancy, Software Freedom. n.d. “Selenium.” Accessed December 30, 2021. https://www.selenium.dev/.