CYO Finnish Covid-19 Confirmed cases forecasting

Sari Inkila

04/08/2021

Contents

1. Executive Summary

1.1 Introduction to the project

This is a report for the Harvard Data Science course series' Capstone -course Choose Your Own Project -assignment. The Choose Your Own -assignment is the last of the two assignments included in the Capstone course.

For this project I chose to analyze the Finnish Covid-19 open data available on the Finnish Institute of Health and Welfare (THL) website. I am aware that the choice was a bit risky considering that we have not covered time series or forecasting in this course series. However, by the time I realized this, I had already invested so much time and effort into downloading and analyzing the data that it seemed easier to just familiarize myself with time series forecasting than finding a new data set and project topic.

I am also aware that to a lot of people the whole topic of Covid-19 might be something they don't want to hear about anymore. However, here in Finland, where we have had it pretty easy up until Christmas 2020, the topic of forecasting the Covid-19 epidemic infection rate is now a hot topic. As a result of our fairly low numbers of confirmed Covid-19 cases compared e.g. to our neighbors Sweden and Russia, we did not have lock downs, mandatory testing on our boarders or mask requirements. In hindsight, this turned out to be a liability as the virus variants started spreading. Now we have the highest number of confirmed Covid-19 cases and hospitalized patients since the beginning of the pandemic.

Our liberal constitution and legal system was not prepared for the need to restrict contacts or enforce mandatory testing on the boarders. The need to change the legislation, and enforce a lock down for the first time since the beginning of the pandemic, has been justified with forecast models created by the Finnish Institute of Health and Welfare (THL) data scientists in co-operation with the local universities. The models referenced by the Finnish government in early March 2021 gave forecasts with exponential growth in Covid-19 infections.

This situation and the need to better understand what the data was telling, got me started on this project topic. I did realize that my knowledge on the field of infectious disease and in time series forecasting is very limited, but I took that as a learning opportunity.

1.2 Data set

This project works with confirmed Finnish Covid-19 case data sourced from the Finnish Institute of Health and Welfare (THL) website

Data sources at THL used were:

- Covid-19 (Corona)data: https://thl.fi/en/web/thlfi-en/statistics/statistical-databases/open-data/confirmed-corona-cases-in-finland-covid-19-
- Online tool data: https://sampo.thl.fi/pivot/prod/en/epirapo/omaolosymp/fact_epirapo_omaolosymp.json
- Vaccination status data: https://sampo.thl.fi/pivot/prod/en/vaccreg/cov19cov/fact_cov19cov.json

The data downloaded includes weekly data points of:

- Population combined across all health care district,
- Number of confirmed deaths resulting from Covid-19,
- Number of confirmed cases of Covid-19,
- Number of people tested for Covid-19,
- Number of symptoms self-reported (in a dedicated online tool screening for need to get tested for Covid-19),

- Number of people referred urgently to be tested (based on the symptoms reported in the screening tool),
- Number of people that have received at least one dose of Covid-19 vaccine.

In addition there was some data related to demographics:

- All confirmed Covid-19 cases per sex and age,
- All Covid-19 deaths per sex and age.

1.3 The goal, model used and the results

The key data for this project was the time series on the weekly confirmed cases of Covid-19. The rest of the downloaded data was only used to better understand the situation. This was due to the openly available data being weekly numbers reported, age group or gender specific total sums. There was no actual individual specific data available.

This meant that there was no way to e.g. predict who might be at higher risk of contracting the disease or dying from it. The only use for the data set was to understand the current and past situation and related statistics, or to try to forecast the future weekly numbers of confirmed Covid-19 cases and/or deaths.

The population of Finland is about 5.5 million and we had it fairly easy in the beginning of the pandemic, so the total death toll of Covid-19 is currently at 860 (Source: Helsingin Sanomat, April 7th, 2021). The number of weekly deaths has not been rising as much as the infections due to vaccinations and risk group isolation recommendations. So, to predict the weekly deaths did not seem to make sense. So, I chose to predict the number of Covid-19 infections based on the available data.

The methods used in the model development were Naïve (from here on just Naive), Simple Exponential smoothing (ses), ARIMA, and Multivariate VAR. All but one of the methods used in this project are univariate using just the past weekly number of confirmed Covid-19 cases. Multivariate VAR (vector autoregression) method was the only one using the other available time series data in addition the time series on confirmed cases. The last model developed was a combination model using the mean point forecasts of the other developed models. As the available data set including the validation data was just 59 weeks, I did not take seasonality into account in the models. Also, tuning of the models was limited.

The goal of this project was to build a forecast model on the development data set (90% of the available data) that forecasts 6 weeks of point forecasts on the validation data set (10 % of the available historical data).

Target of the project was to build a point forecast model. The final result achieved with the best model using the Naive method resulted in time series cross-validation RMSE of 620 on the validation set. The six weeks point forecast was about 1750 cases lower than the observed values in the validation data set. In other words, the forecast did not perform well.

1.4 Key steps

As time series forecasting was not part of the Harvard Data Science course series curriculum I had to do extra steps to understand what to do. This included a lot of online research on both the forecast methods in general as well as the R libraries and functions required. I had to also gain an understanding of the performance metrics, prediction intervals and method accuracy evaluation.

The key steps that were performed:

1. Identifying interesting data sets openly available
2. Defining the problem: forecasting Covid-19 infections 6 weeks into the future (on the validation data set)
3. Conducting preliminary analysis including plotting the data to understand potential seasonality, trends and patterns as well as outliers
4. Choosing and fitting models

- Starting with gaining a better understanding of time series and forecasting models related to time series
- Forecasting models chosen were:
  - A baseline using the Naive method
  - Simple exponential smoothing method
  - Auto ARIMA method
  - Multivariate vector autoregression (VAR) method
  - Combined model using mean point forecast of the other models

5. Using and evaluating forecasting models

- Validation of the developed model on the validation data set and calculating performance metrics of the model on the validation data set.

---

2. The Analysis

2.1 Data Cleaning

The data was downloaded from the Finnish Institute of Health and Welfare (THL) website using their THL json interface. The interface required use of parameters to extract the different data sets.

Data sources at THL used were:

- Covid-19 (Corona)data: https://thl.fi/en/web/thlfi-en/statistics/statistical-databases/open-data/confirmed-corona-cases-in-finland-covid-19-
- Online tool data: https://sampo.thl.fi/pivot/prod/en/epirapo/omaolosymp/fact_epirapo_omaolosymp.json
- Vaccination data: https://sampo.thl.fi/pivot/prod/en/vaccreg/cov19cov/fact_cov19cov.json

Data downloaded from the Covid-19 (Corona) data interface were:

- Population combined across all health care district,
- Number of confirmed deaths resulting from Covid-19,
- Number of confirmed cases of Covid-19,
- Number of people tested for Covid-19,
- All confirmed Covid-19 cases per sex and age,
- All Covid-19 deaths per sex and age.

Data downloaded from the Online tool data interface were:

- Number of symptoms self-reported (in a dedicated online tool screening for need to get tested for Covid-19),
- Number of people referred urgently to be tested (based on the symptoms reported in the screening tool).

Data downloaded from the Vaccination data interface were:

- Number of people that have received at least one dose of Covid-19 vaccine.

The data was then joined to following data frames

- One containing the weekly sums time series data for each week in the periods starting from week 8/2020 to 12/2021,
- One containing the age group specific data (confirmed cases and deaths),
- One containing the gender group specific data (confirmed cases and deaths).

There were some anomalies in the data. E.g. weekly population was available only for 2020, so the 2021 weekly numbers were taken from the THL json interface for the vaccination data as it was the more recent data set. The date format used by the THL interface required some modification to get it into a datetime format required by the time series processing. ISOweek2date() was used to get the week number and corresponding Monday of each week to match and to correctly include the week 53 of 2020.

2.2. Data visualization and scaling

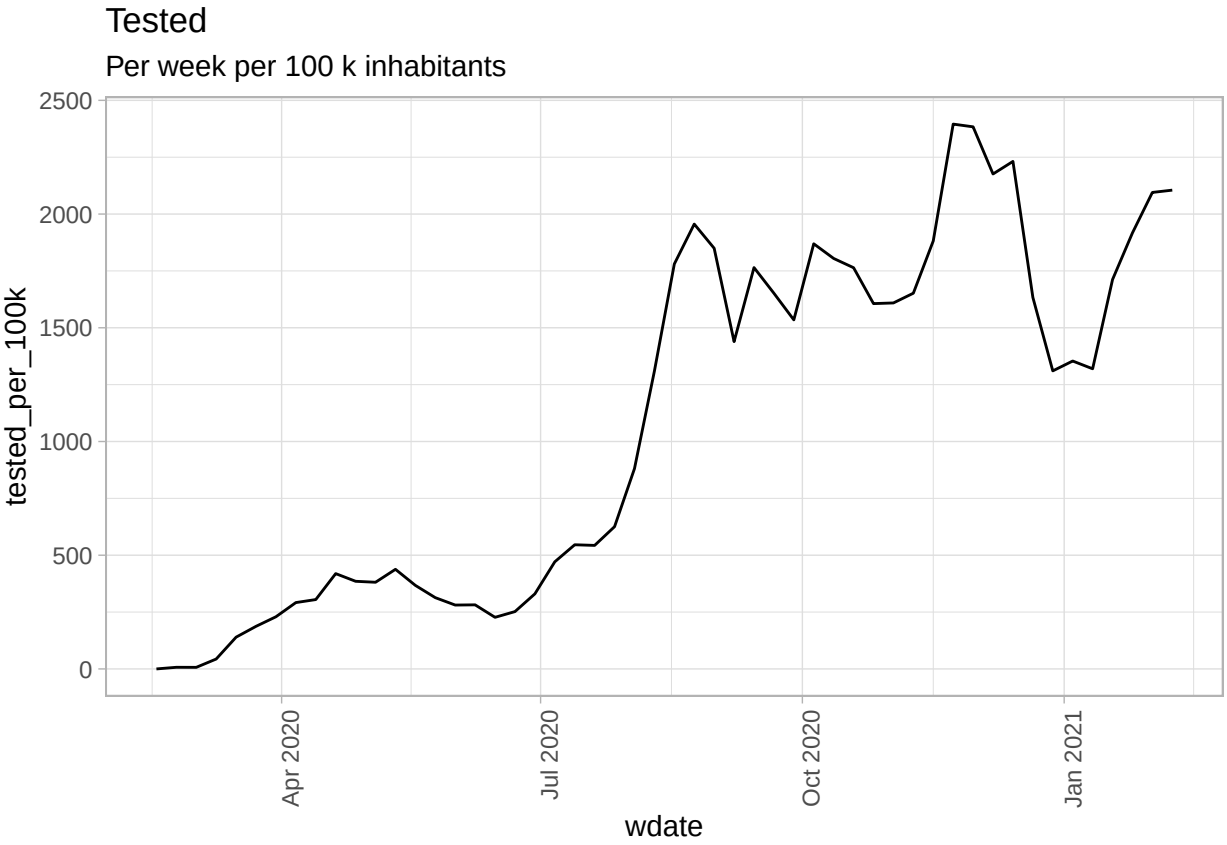
In order to better understand the data I created metrics commonly used with Covid-19 data i.e.

- scaling the weekly numbers per 100 000 inhabitants,

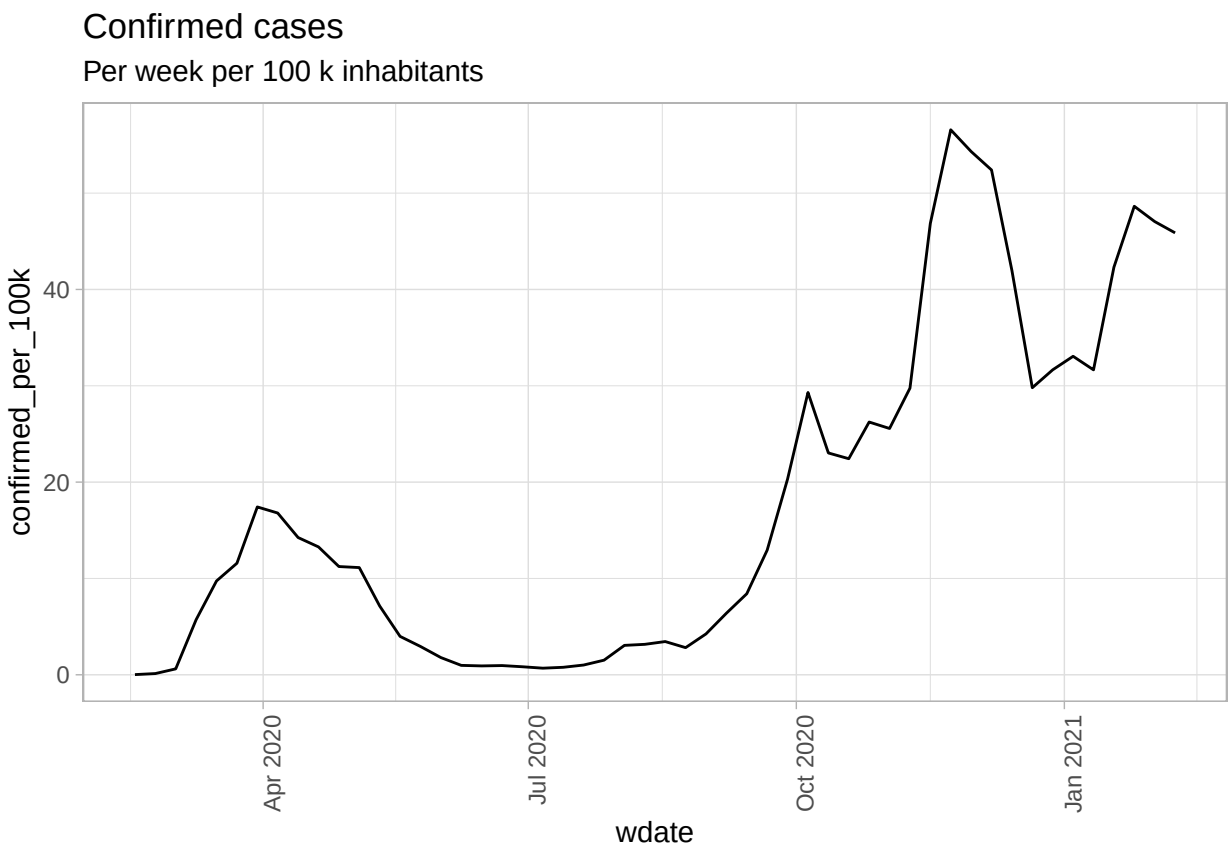
- calculating the case fatality rate (the number of COVID-19 deaths in a population / Divided by the total number of COVID-19 cases * 100),
- mortality rate (the Covid-19 deaths per 1000 inhabitants in a year) for the year 2020.

The plots below visualize the tested, confirmed cases and deaths per 100 000 inhabitants up until the week 6, 2021.

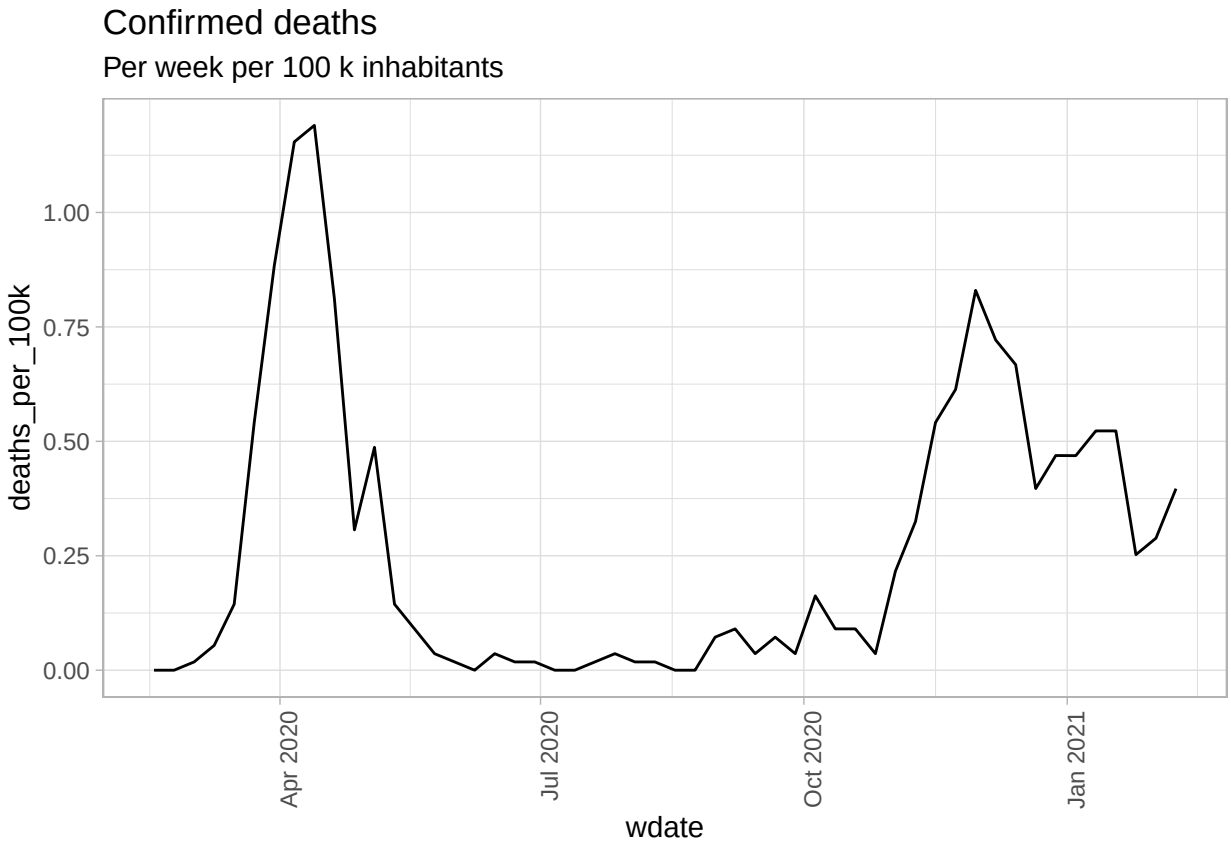
The below plot visualizes the weekly number of tested per 100 000 inhabitants.



The below plot visualizes the weekly number of confirmed cases per 100 000 inhabitants.

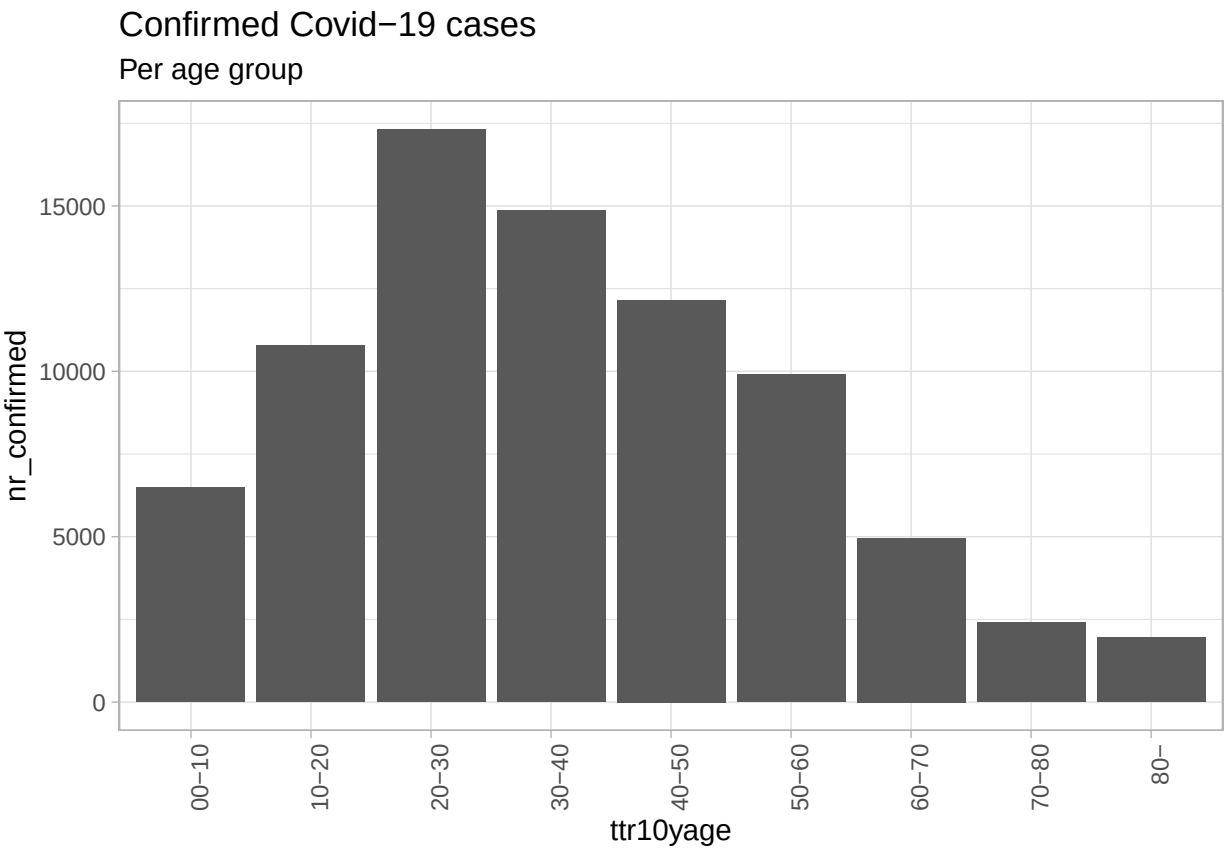


The below plot visualizes the weekly number of deaths per 100 000 inhabitants.

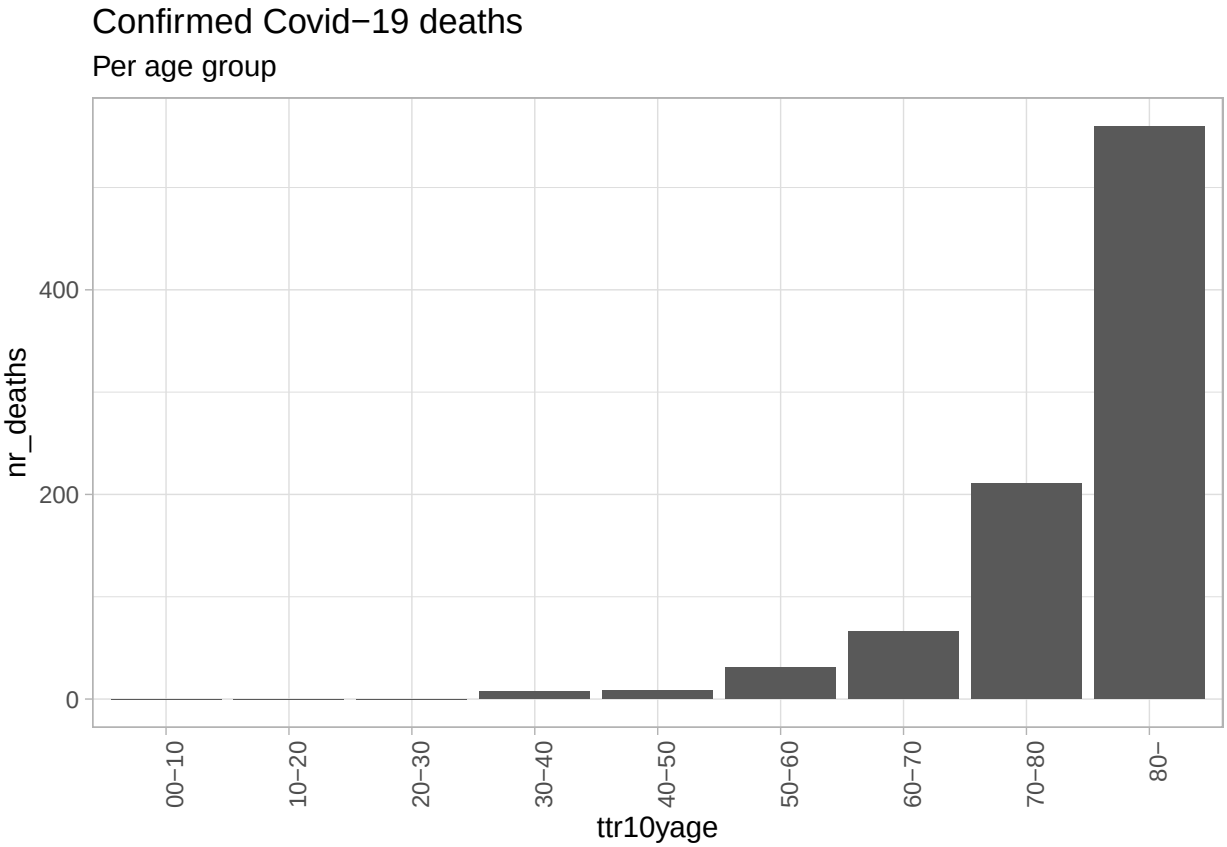


The plots below illustrate the number of confirmed cases of Covid-19, the related deaths and case fatality rate per age group.

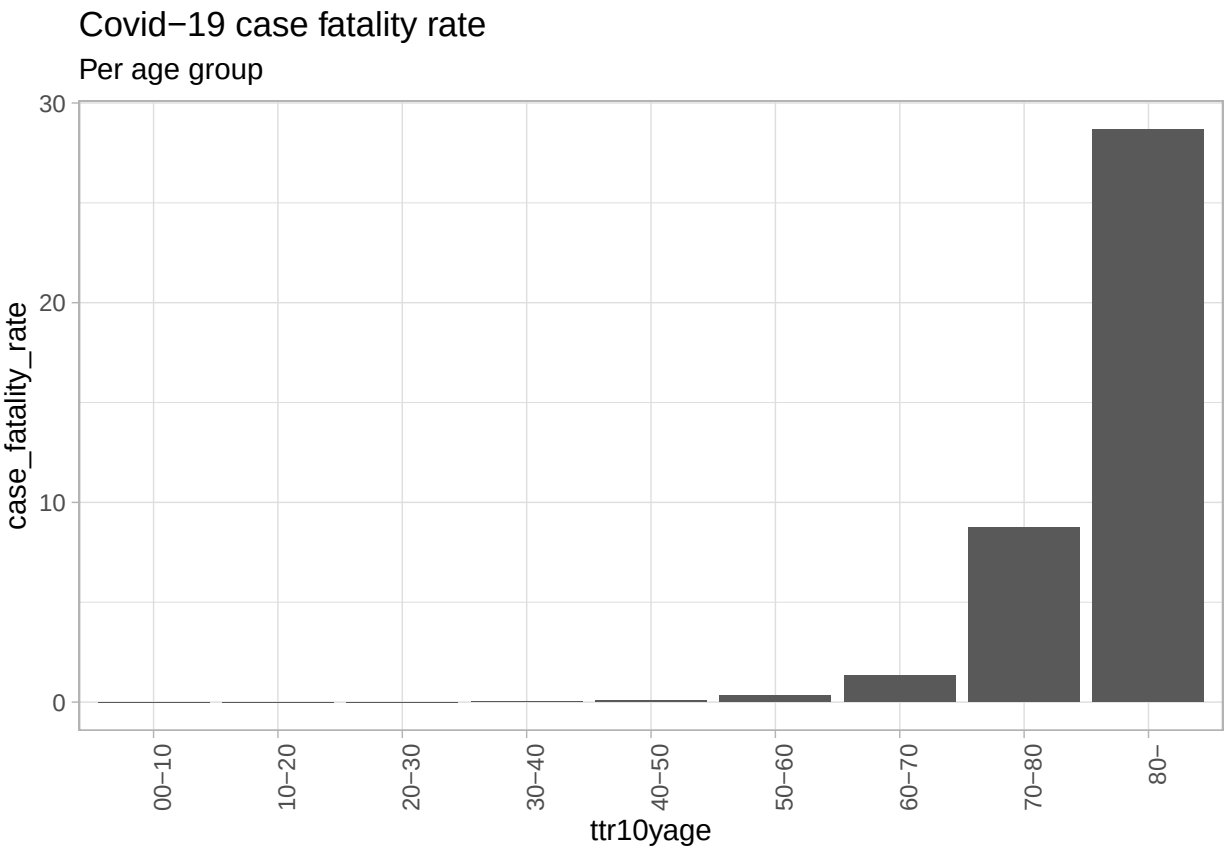
The age distribution of the number of confirmed cases:



The age distribution of the number of deaths:

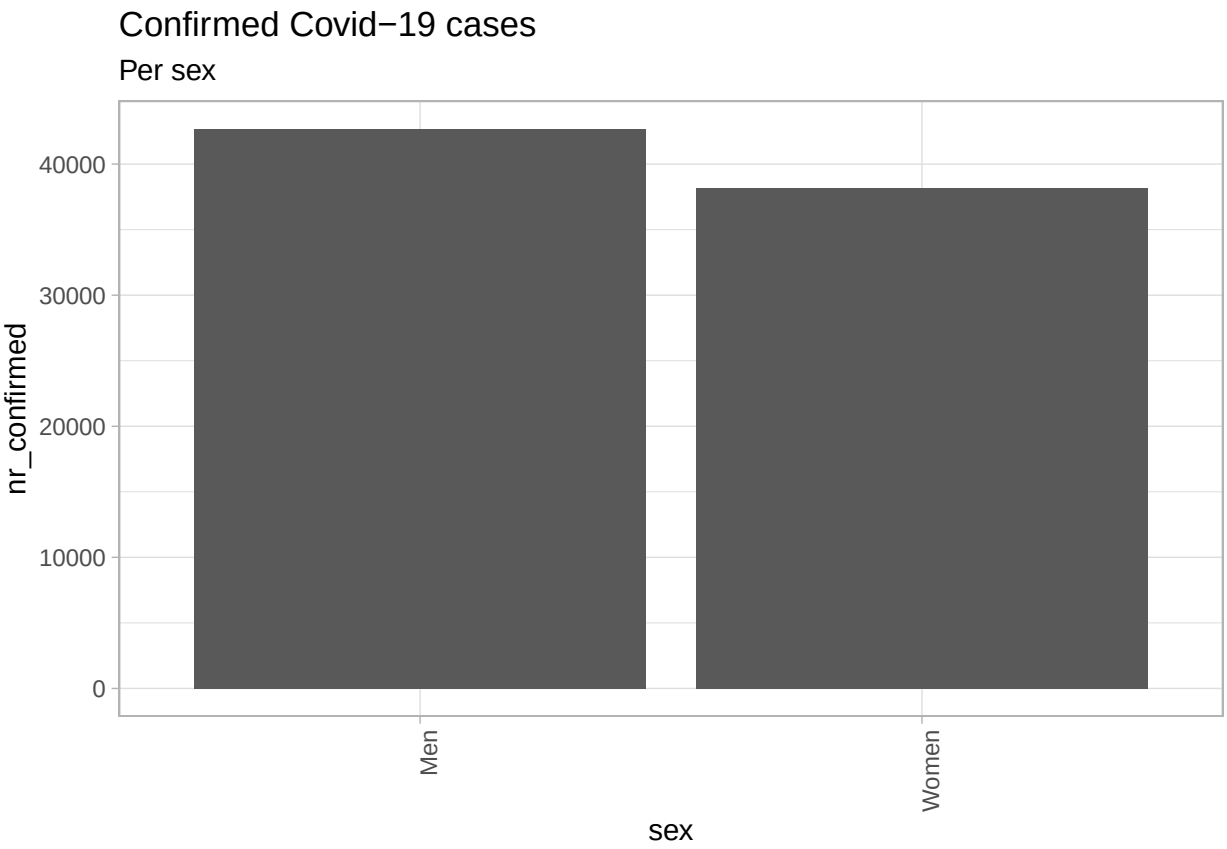


The age distribution of the number of case fatality rate:

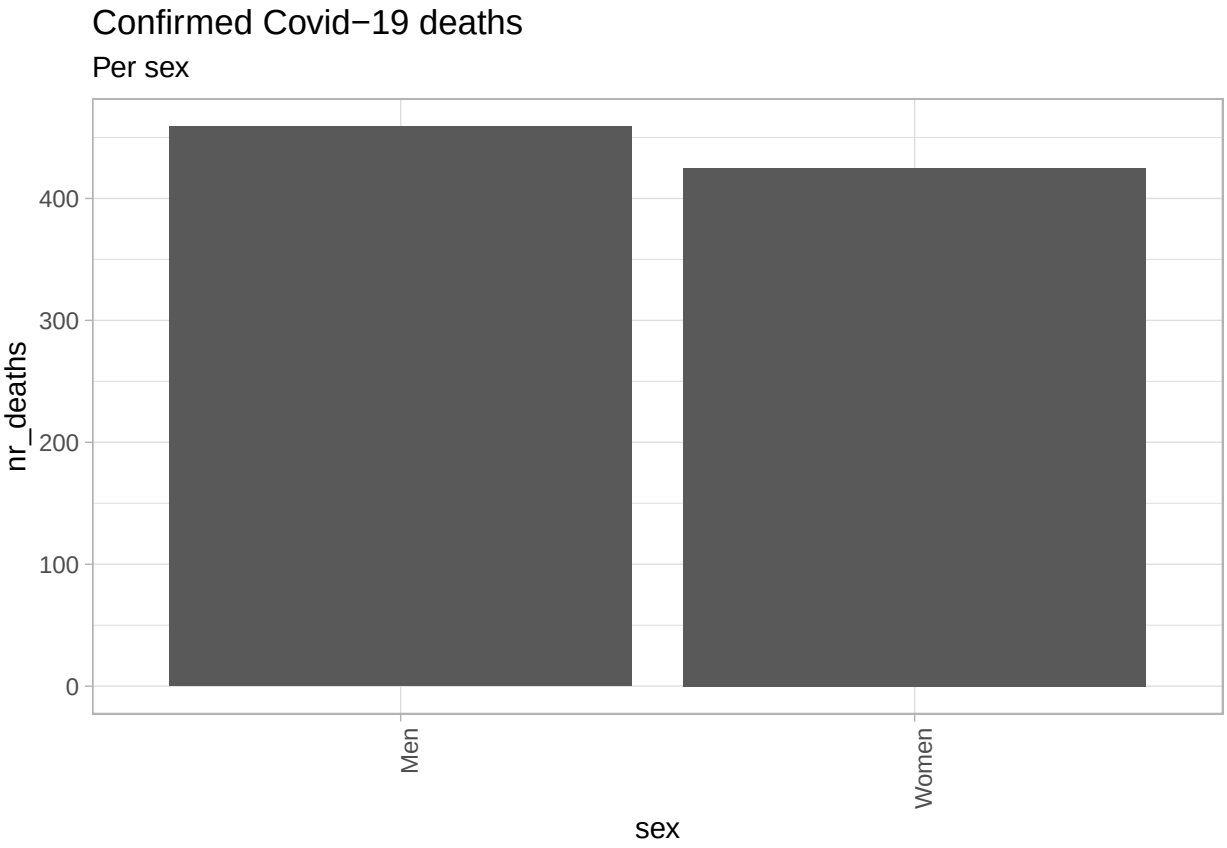


The plots below illustrate the number of confirmed cases of Covid-19, the related deaths and case fatality rate per age gender.

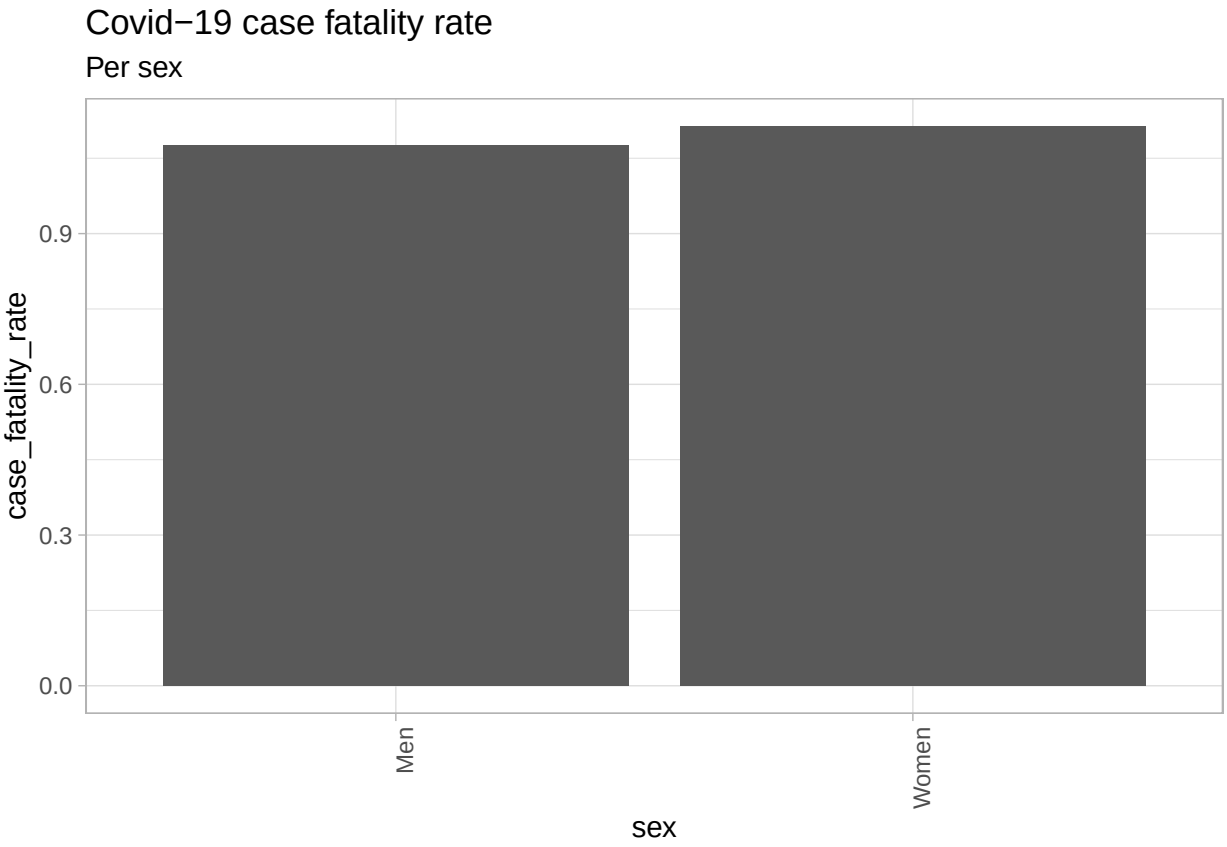
The gender distribution of the number of confirmed cases:



The gender distribution of the number of deaths:



The gender distribution of the case fatality rate:



2.3 Creating the training and test sets as time series for model development

As the Covid-19 data was time series data and there was a limited amount of it available, the data was split into:

- a development data set: the data up to week 6 / 2021 (90%),
- a validation set (=final hold-out test set): the last 10% of the available data (weeks 7-12/2021).

For the forecast development the development data set was split into a separate training and test sets:

- Train set: the first 41 weeks (80% of the development data set),
- Test set: the last 11 weeks (20% of the development data set).

2.4 The model development

2.4.1 Model performance metrics for forecasting

Determining the model performance using:

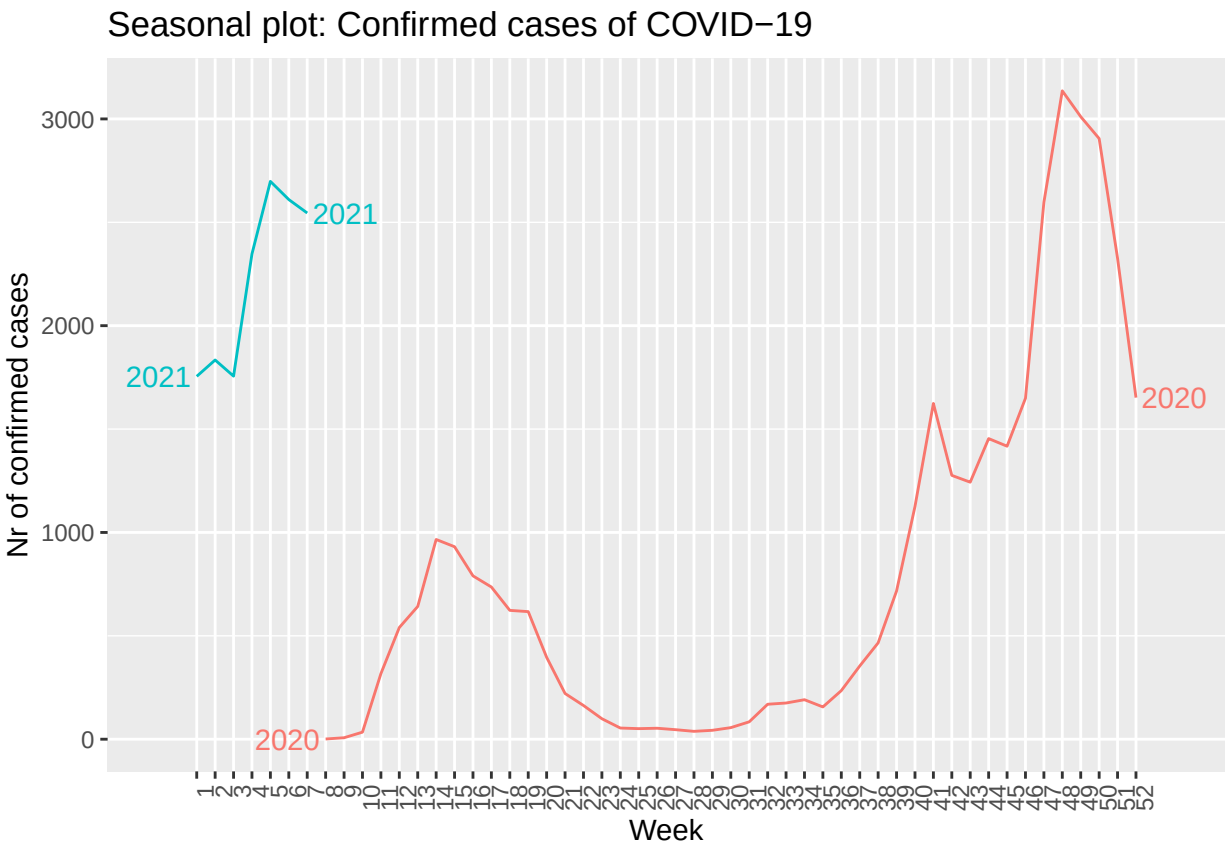
- RMSE (a residual mean squared error)
- Mean Absolute Error (MAE)
- Mean absolute percentage error (MAPE)
- RMSE on time series cross-validation.

The theory on the forecasting metrics states the following:

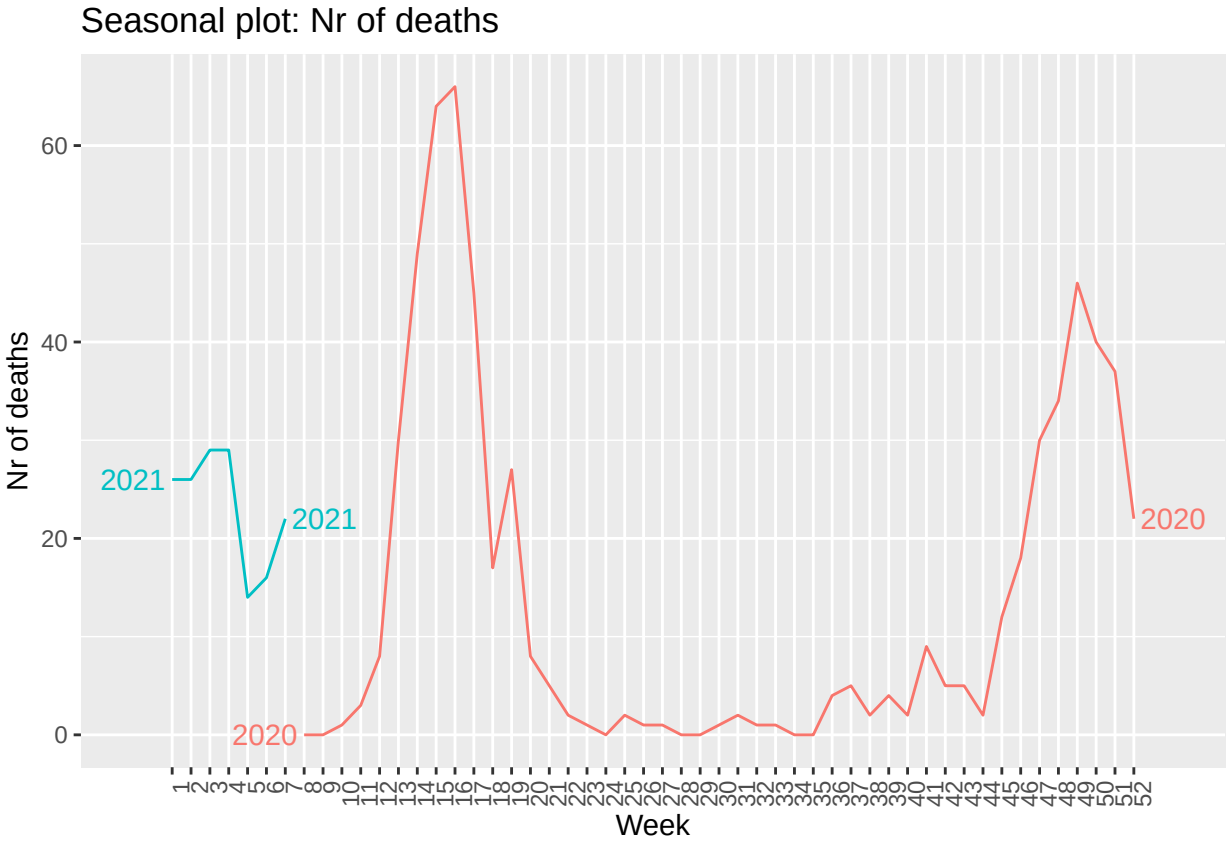
> "A forecast method that minimizes the MAE will lead to forecasts of the median, while minimizing the RMSE will lead to forecasts of the mean. Note that the the forecast errors MAE and RMSE are on the same scale as the data. Percentage errors (MAPE) have the advantage of being unit-free, and so are frequently used to compare forecast performances between data sets. However, measures based on percentage errors have the disadvantage of being infinite or undefined if training data observation in the period of interest are zero, have extreme values or are close to zero. Another problem with percentage errors that is often overlooked is that they assume the unit of measurement has a meaningful zero. A good way to choose the best forecasting model is to find the model with the smallest RMSE computed using time series cross-validation." (Source: https://otexts.com/fpp2/accuracy.html)

2.4.2 Preliminary time series analysis

Time series have few new dedicated plot types that visualize the time series specific qualities e.g. seasonality. Below you can see the ggseasonplot() plots for both confirmed Covid-19 cases and deaths.



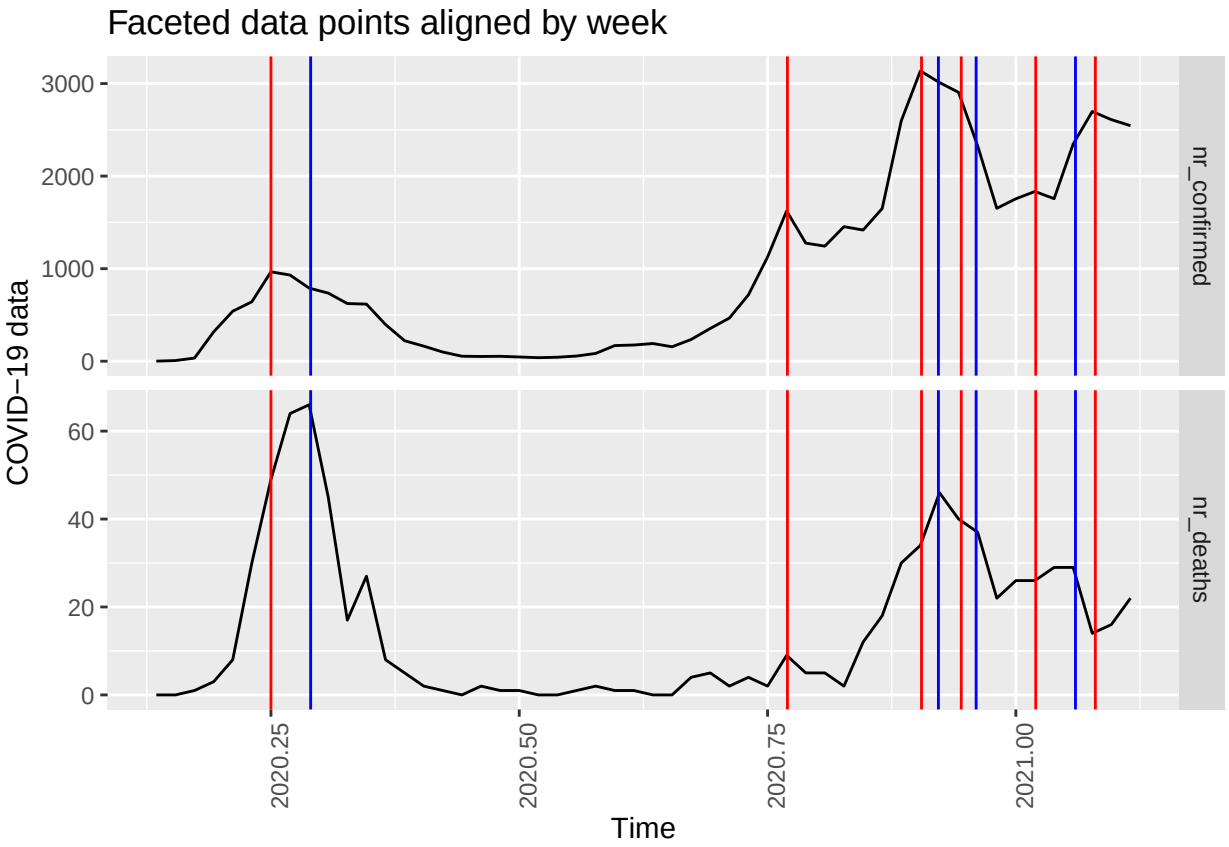
The seasonal plot above indicates that there is alikely seasonality in the data, however, with only 1 year of data it can't easily be used in the forecasts. Also, we need to keep in mind that the first half of 2020 there was not enough testing capacity to confirm all of the Covid-19 cases.



This is suggested also by the seasonal plot of the Covid-19 related deaths that has a spike starting with the week 11/2020 until week 18, but the size of the corresponding peak in the confirmed cases leading up to those weeks in

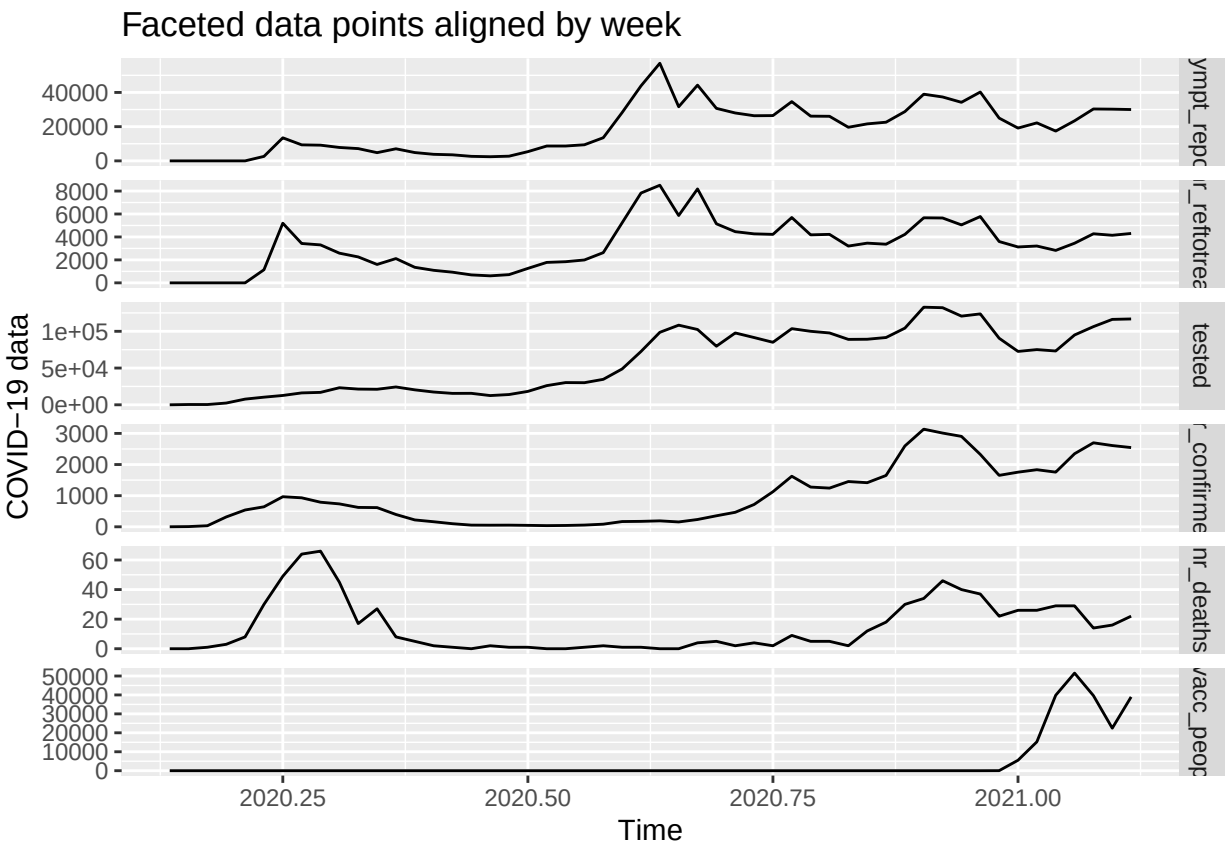
2020 was not proportional. At that time there was not enough testing capacity.

If you look at the plot below, where I have faceted the two data sets and marked the peaks in confirmed cases in red and peaks in deaths in blue, the size of the initial peak in confirmed cases is lower than you would expect when contrasting with the corresponding peaks in the deaths in the latter part of the data.

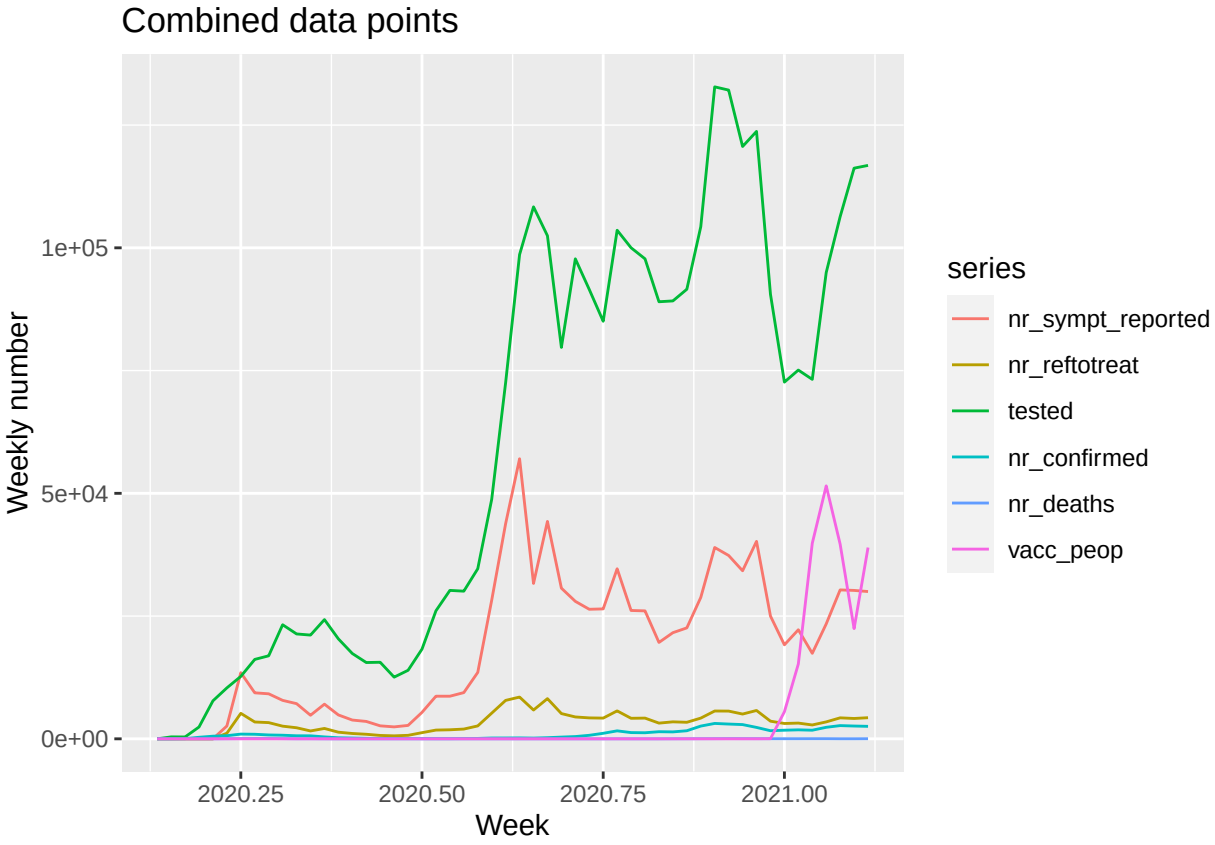


Similar lack of accurate data is present in the other data points that are faceted in the plot below. The initial peak in the number of Covid-19 deaths is greater than any of the latter peaks, however, none of the other data points have similar initial peaks in proportion to the deaths.

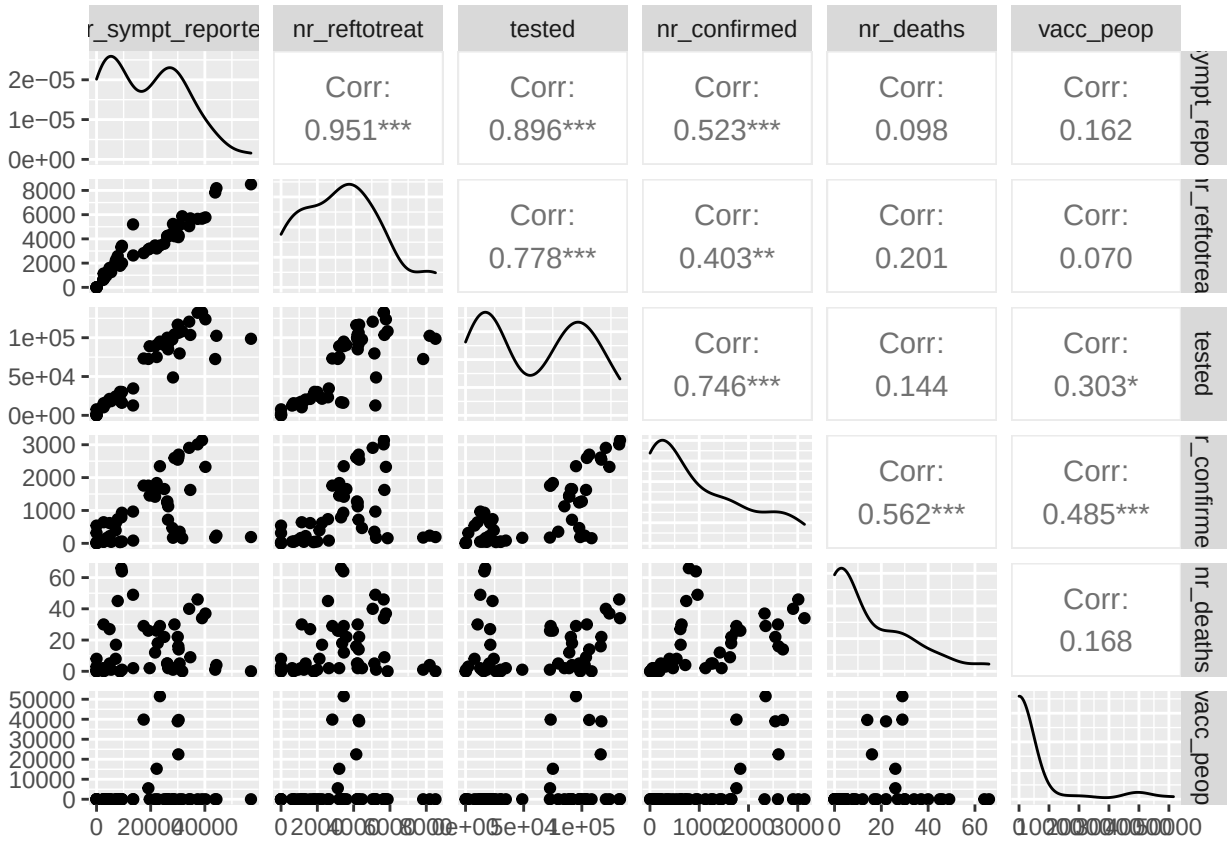
The tools for pre-screening people for testing in order of highest priority, where developed early, but the usage was a little slow to pick up. This would suggest that the related time series data that I expected to be leading indicators (Number of people tested, Number of symptoms self-reported and Number of people referred urgently to be tested) have only limited ability to support any prediction or forecasting efforts.



When we combine the different time series into a one plot we can see that the usage of the tool (used for screening for testing) never picked up to extend that it would have volumes higher than the number of people that were actually tested. This is another reason that the data sets are not as useful as I hoped for.



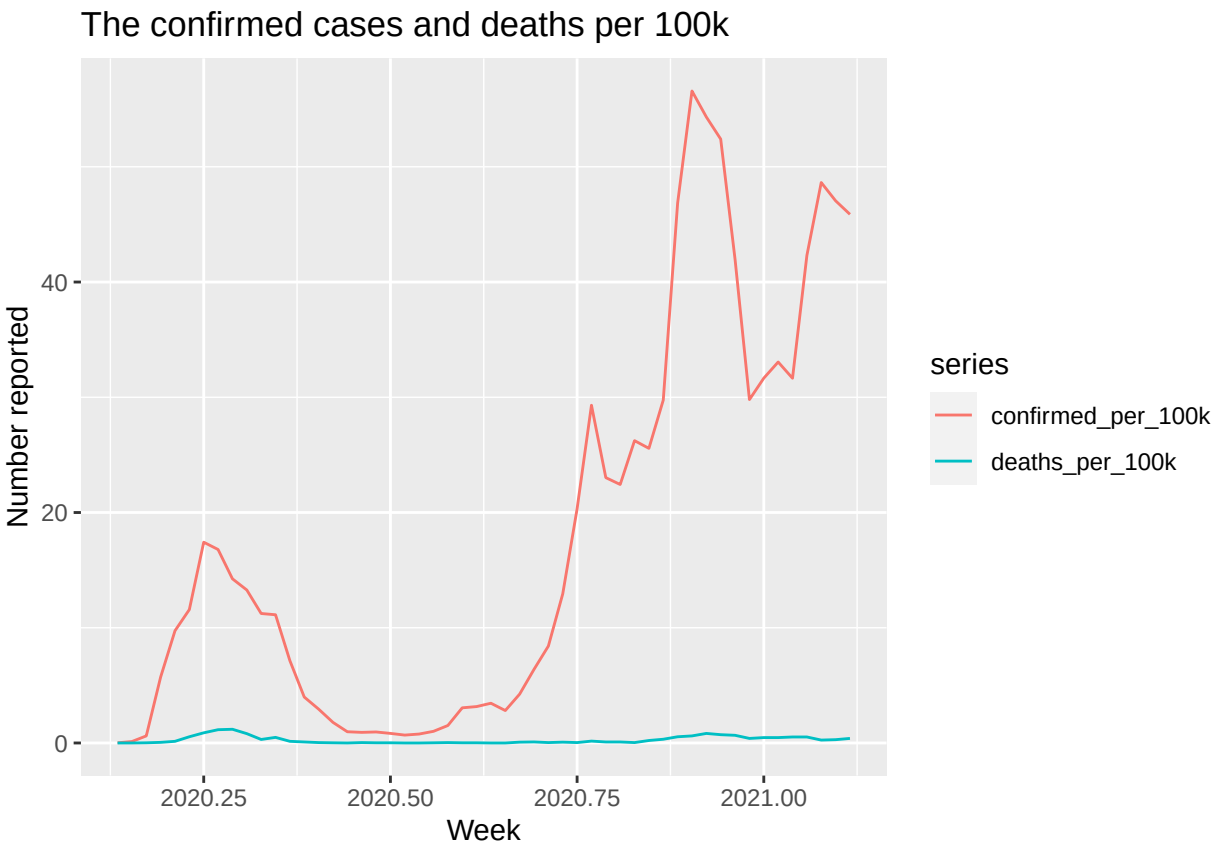
To further understand the relationship of the time series data, I ran the GGally::ggpairs generalized pairs plot that provides a matrix with both plots and numeric correlations. You can see the results below:



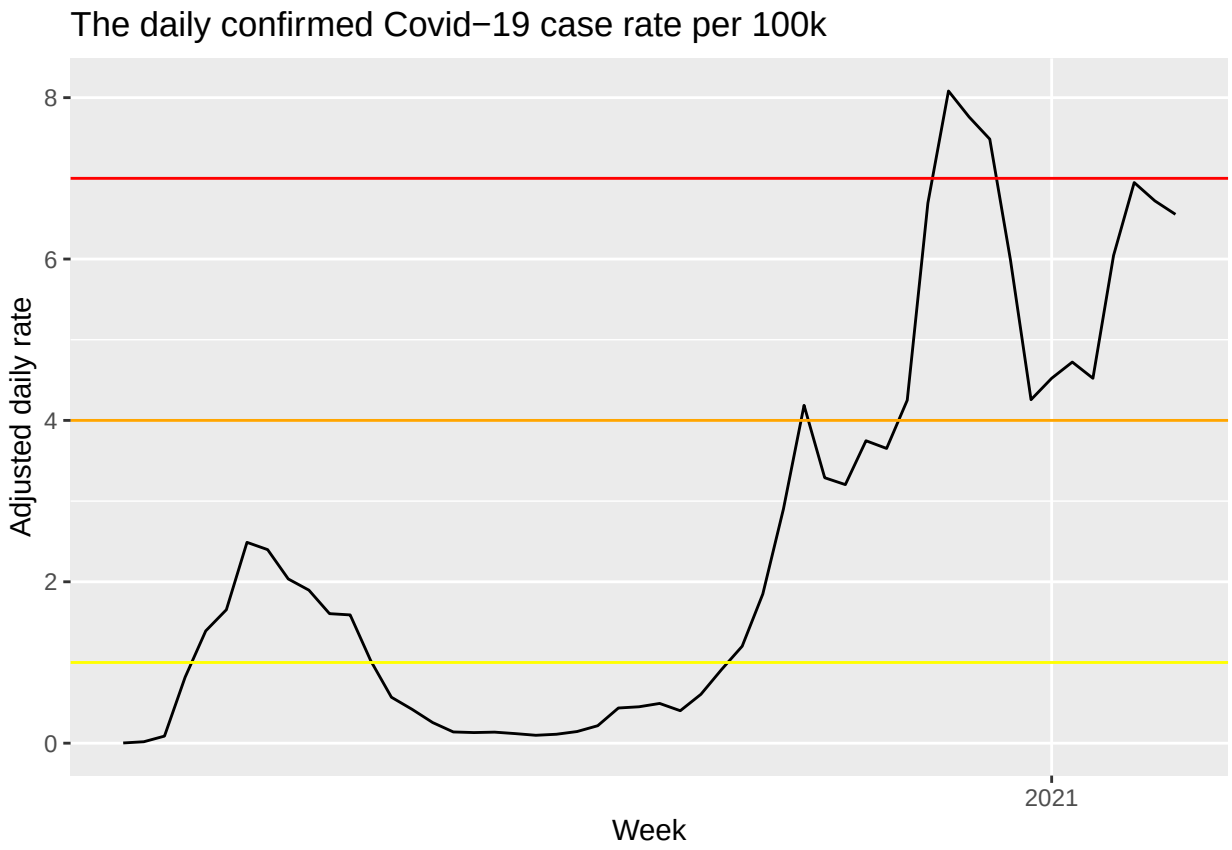
The conclusions from the matrix:

- As the first three time series data are used to determine who should be tested it is to be expected that they have a fairly high correlation.
- Number of people vaccinated is so low and has been on-going for such a short time, that it is not going to have much impact on anything yet. For this reason it is not used in the model development.

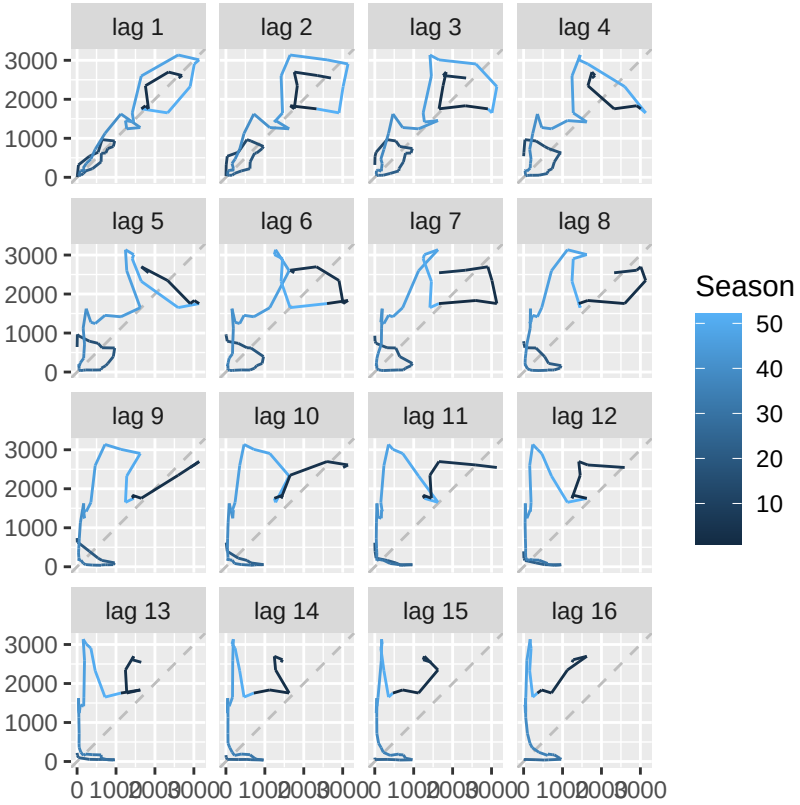
When we look only at the weekly confirmed cases and deaths per 100 000 inhabitants, we can see that the number of deaths remains so low that forecasting it in this limited data set would be difficult. This is the reason why this project focuses on the point forecast of the confirmed Covid-19 cases.



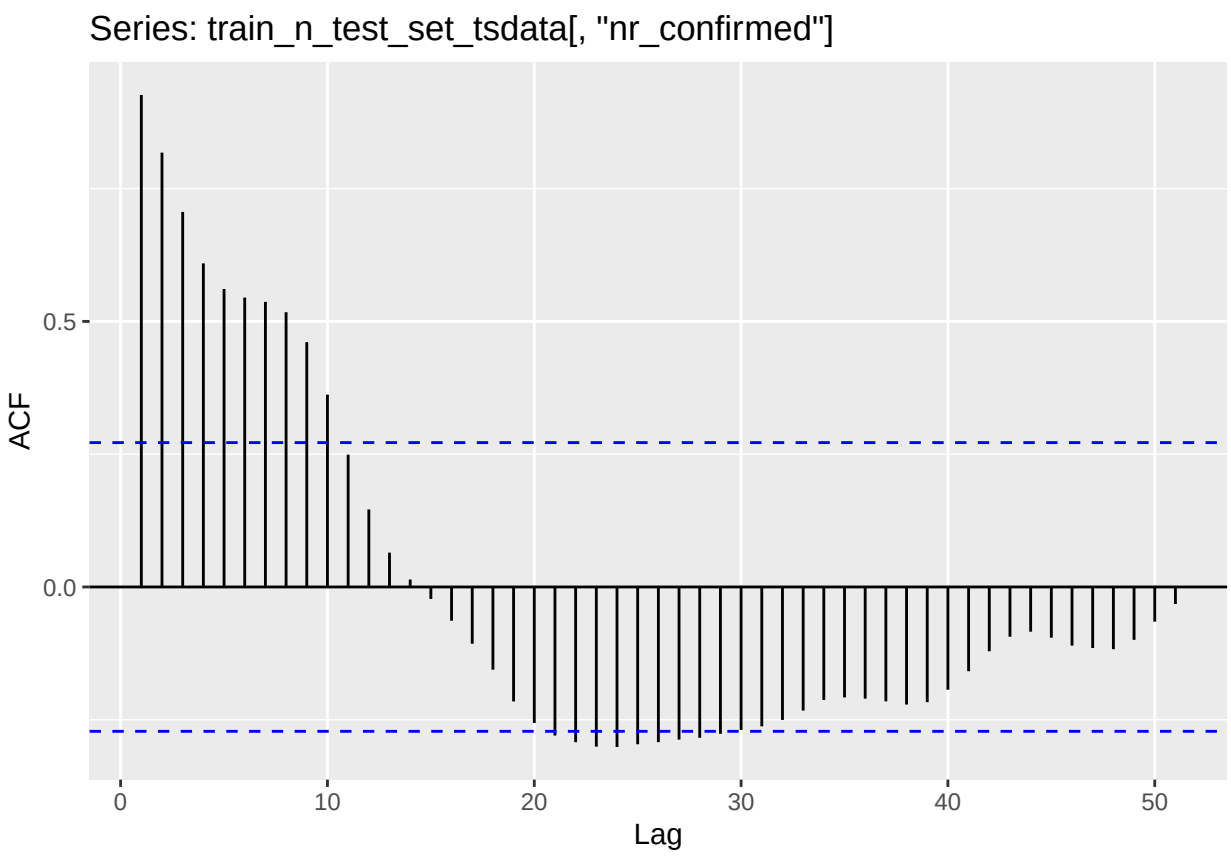
As said the epidemic stayed at a fairly low level for most of 2020 and only recently reached tier labeled widespread (over 7 cases with the metric of seven-day average for cases per 100 000 inhabitants). As you can see from the plot below, the first time the threshold was surpassed was when the first cases of the mutated variants started spreading in Finland around November and December 2020. But after that the case numbers appeared to go down during the holiday season (around Christmas and New Year). The cut-off point between the development and data set is right at the point, when the case numbers have started to decline again, but are under the highest threshold.



Gglagplot() was used to gain an understanding of the possible autocorrelation of the confirmed Covid-19 cases time series. Using the gglagplot we can see that the highest autocorrelation is with lag 1.



We can confirm this with the ggAcf(), which shows results of the autorrelation. There was a significant autocorrelation up to 11 lags, but the first lag is the highest.



The theory states:

> "When data has a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. Which means that the ACF of trended time series tend to have positive values that slowly decrease as the lags increase." (source: https://otexts.com/fpp2/autocorrelation.html)

According to this theory, I can conclude that the confirmed cases data appears to have a trend.

These plots together show that there are challenges to be expected in developing the forecasting models:

- Data appears to have seasonality, but there is only data for one year. So it will be difficult to take it into account.
- Data is incomplete as there was not enough testing capacity at the beginning of the pandemic and only the severe cases got tested.

To a novice creating time series forecast model with this data set will be a real challeng.

2.4.3 Forecasts with Naive Method

To get started I created a baseline forecast on naive(), which forecasts the last observed value. The definition of naive method is that we simply set all forecasts to be the value of the last observation. The forecast was created for the 11 weeks following the time series in the training set and resulting point forecast was tested against the test set actual data for the same 11 weeks.

To determine the best fit forecast method I ran the time series cross-validation on the development test set (including training and test sets) and checked the residuals of the point forecast based on the training set. After that I calculated the accuracy of the training set forecast using the 11 weeks test set. All model performance metrics were saved to a data frame to enable comparison between the models.

According to theory:

> "Residuals are useful in checking whether a model has adequately captured the information in the data.

A good forecasting method will yield residuals with the following properties:
* The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.
* The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts."

"In addition to these essential properties, it is useful (but not necessary) for the residuals to also have the following two properties.
* The residuals have constant variance.
* The residuals are normally distributed." (Source: https://otexts.com/fpp2/residuals.html)

The first check of the residuals was the mean of residuals: [1] 78.375

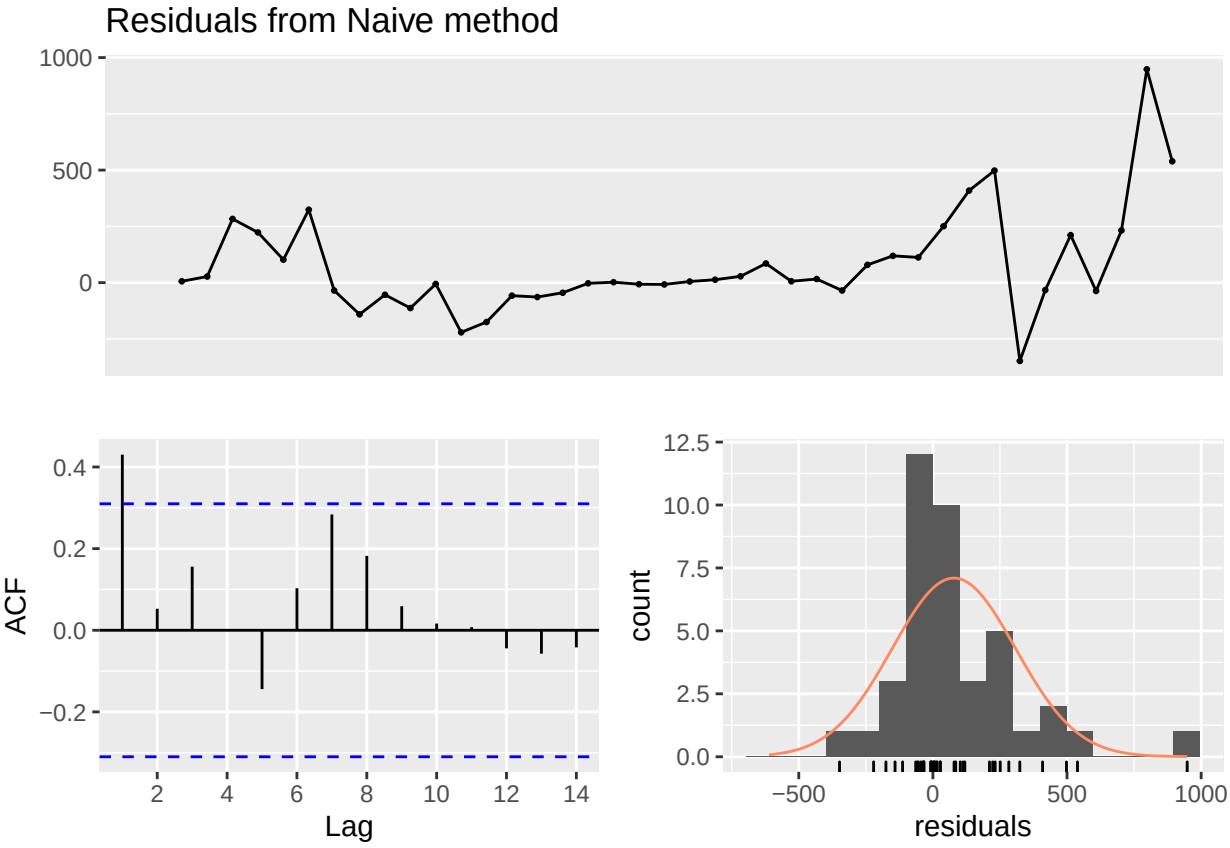
Which is not zero and would indicate that there is a bias.

Next I checked the residuals with checkresiduals() which produces:

- a time plot,
- ACF plot and histogram of the residuals (with an overlaid normal distribution for comparison)
- a Ljung-Box test with the correct degrees of freedom.

For the use of the Ljung-Box test the theory states:

"Ideally, we would like to fail to reject the null hypothesis. That is, we would like to see the p-value of the test be greater than 0.05 because this means the residuals for our time series model are independent, which is often an assumption we make when creating a model." (Source: https://www.statology.org/ljung-box-test/)



Ljung-Box test

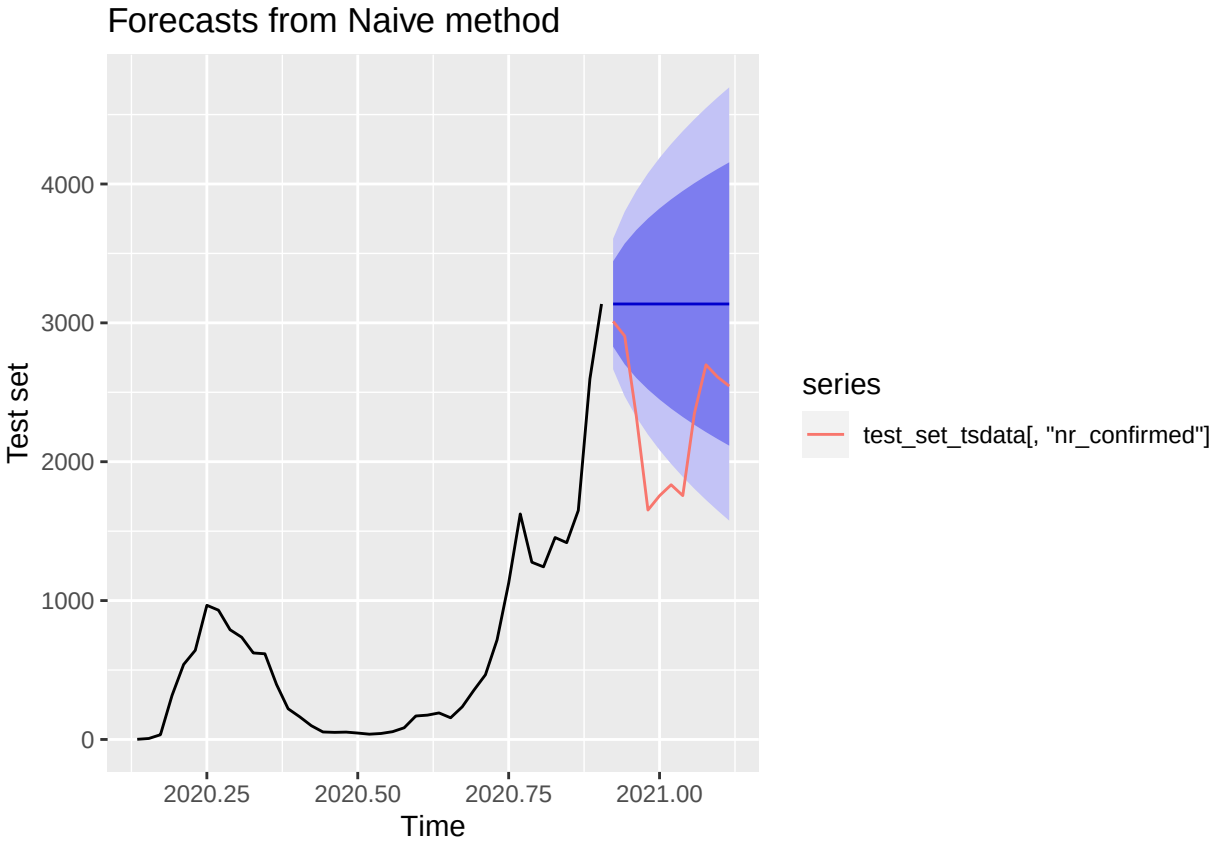
data: Residuals from Naive method Q* = 16.56, df = 8, p-value = 0.03503

Model df: 0. Total lags used: 8

Conclusion:

- The variations in the residuals do not stay the same over time,
- ACF of residuals suggests that naive model hasn't captured all the elements affecting the forecast,
- Residuals do not appear to be normally distributed,
- from Ljung-Box test with a small p-value (under 0,05). Thus, the null hypothesis of the test holds and we conclude that the data values are not independent i.e. the residuals are not just white noise.

I used autoplot() to visualize the point forecast results and compared them to the test set actuals. The red line is the actual test set data, black line is the mean point forecast and the fan shows the 95% and 80% prediction intervals.



The conclusion from visual inspection of the plot was that there were elements left to use in forecasting that the naive model did not capture. The observed values were not in the point forecast's 95% prediction interval. However, the last observed values of the test set are within the 95% prediction interval and closing on the point forecast.

I checked the forecast errors, i.e. the difference between an observed value and its forecast. The forecast errors were calculated against the test set.The last column in the table (RMSE_CV) was produced by the cross-validation using the tsCV().

method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Naive	Test set	947.4573	823.3636	41.65945	756.7302

The table above with the results of the accuracy metrics show that the Naive method did not capture all the elements affecting the forecast. Accuracy of the method using mean average error (MAE) are off by [1] 823.3636 confirmed cases.

2.4.4 Forecasts with Simple Exponential Smoothing Method

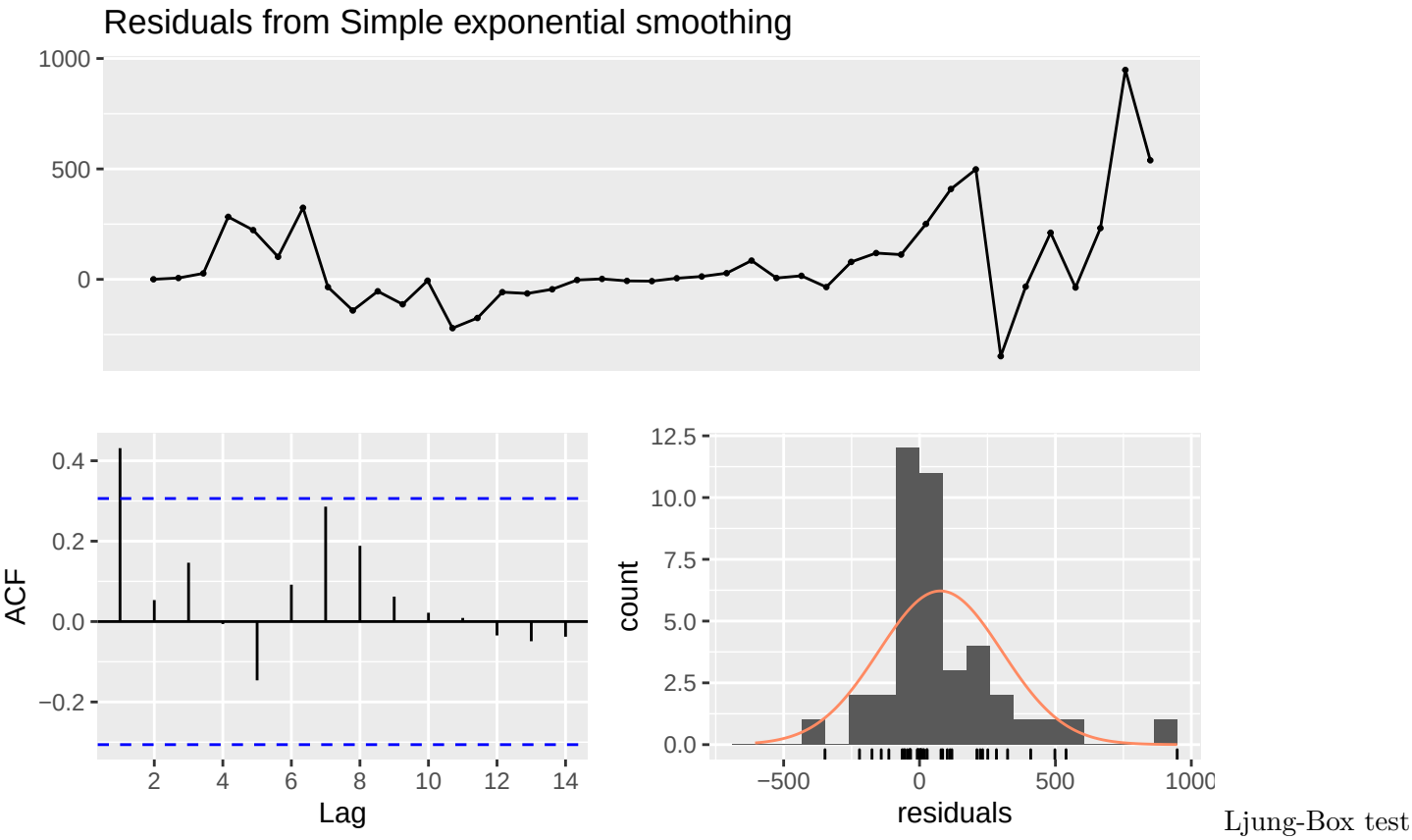
According to theory:

> Exponential smoothing methods are an extension of the naive method, where the forecasts are produced not with the mean of the past values, but with the weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the recent values get higher weights and vice versa. (Source: https://www.pluralsight.com/guides/time-series-forecasting-using-r)

Again I first checked the mean of residuals: [1] 76.47633

The mean of the residuals is not zero indicating that the ses() model has a bias.

Next the I checked the checkresiduals().



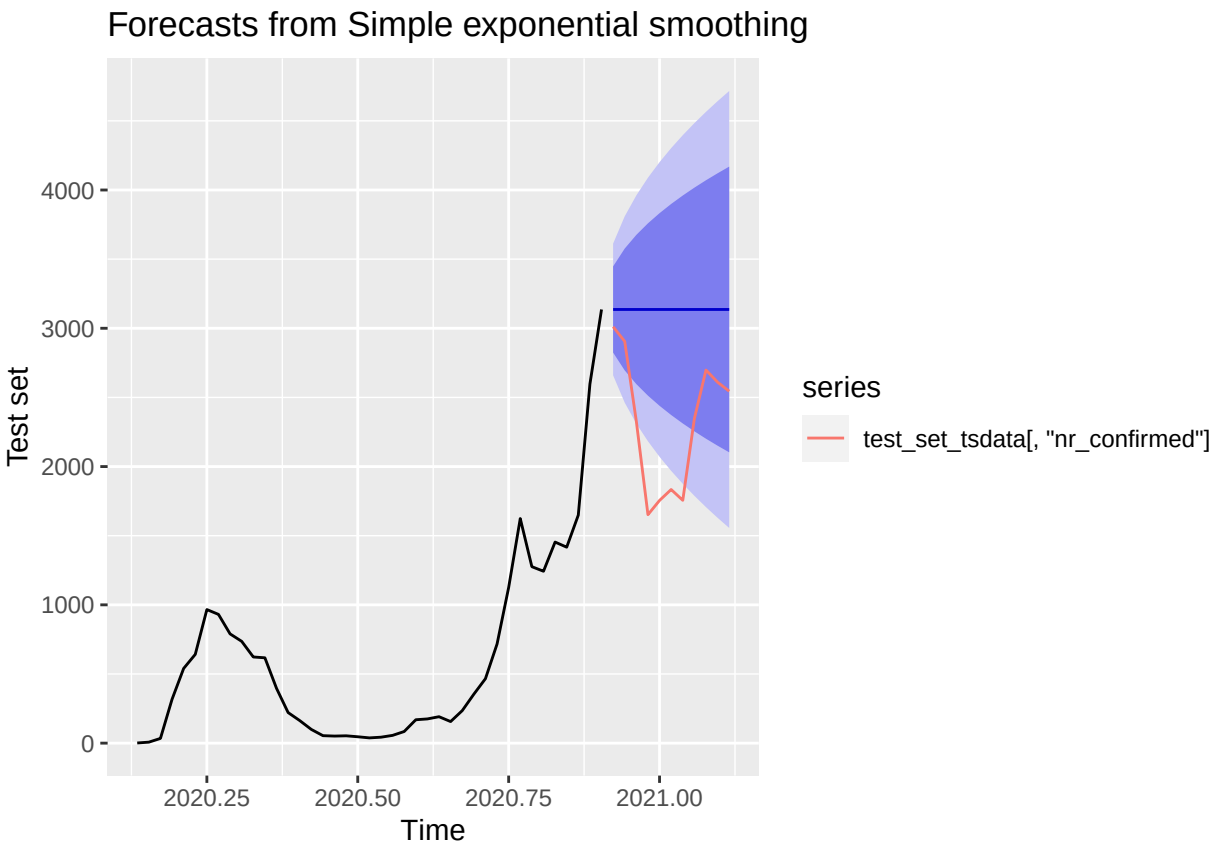
Ljung-Box test

data: Residuals from Simple exponential smoothing Q* = 16.929, df = 6, p-value = 0.009549

Model df: 2. Total lags used: 8

Conclusion:
* The variations in the residuals do not stay the same over time,
* ACF of residuals suggests that naive model has not captured all the elements affecting the forecast
* Residuals do not appear to be normally distributed,
* from Ljung-Box test with a small p-value (under 0,05). Thus, the null hypothesis of the test holds and we conclude that the data values are not independent, i.e. the residuals are not just white noise.

Below plot shows the visualization of the forecast results compared to the test set:



The visualization of the simple exponential smoothing looks very similar to the naive model.

The difference between an observed value and its point forecast was checked using accuracy(), where the forecast errors were calculated against the test set.

method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Simple exponential smoothing	Test set	947.4104	823.3097	41.65701	756.8346
Naive	Test set	947.4573	823.3636	41.65945	756.7302

Here we can see that the results of the accuracy metrics are almost identical to the Naive method, but the cross-validation RMSE was actually tiny bit better with the Naive model.

2.4.5 Forecasts with Auto ARIMA Method

Researching online for models used in times series, and in COVID-19 time series in particular, one of the basic models used was ARIMA. (Reference: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0244173)

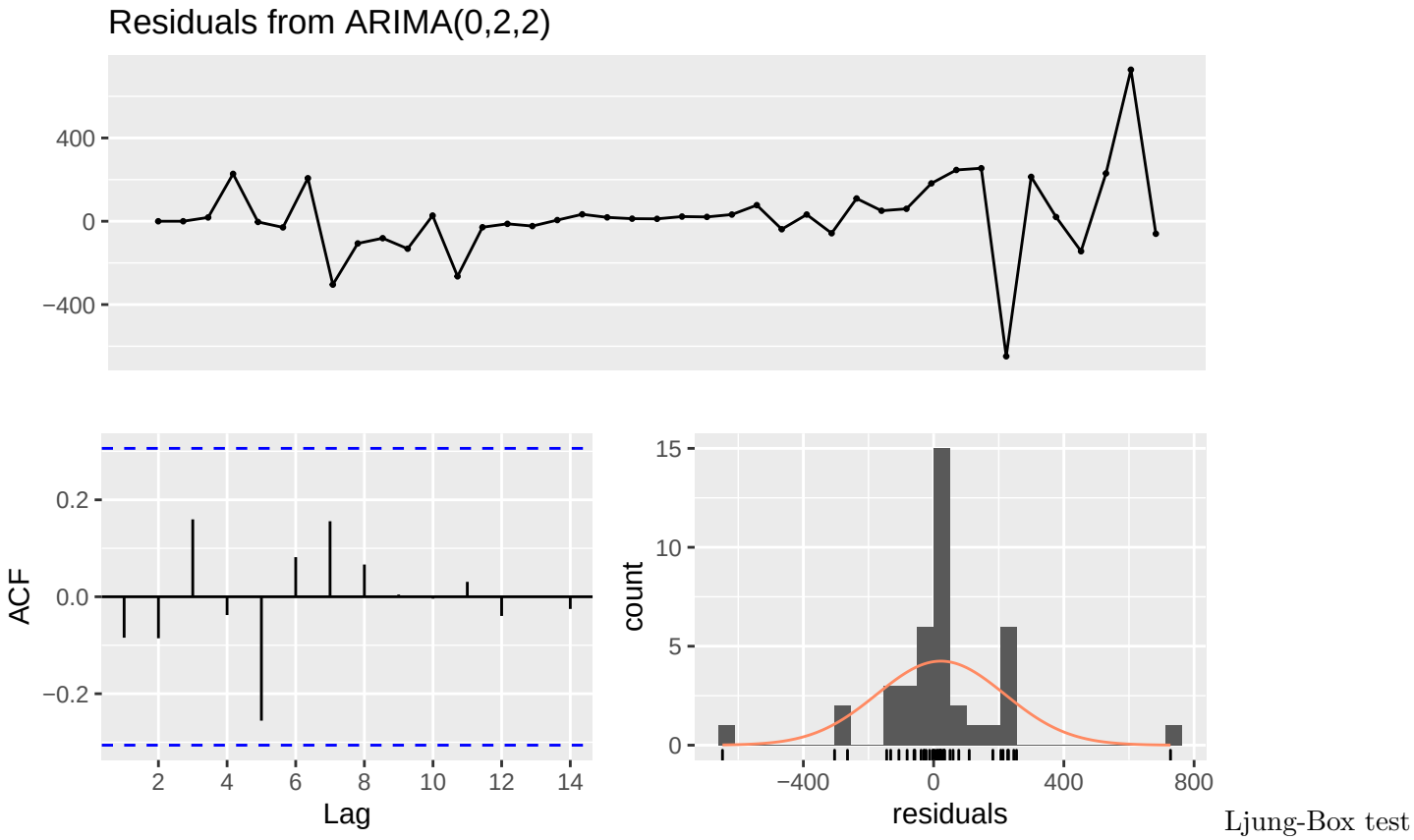
The theoretical definition of the ARIMA:

> "When differencing is combined with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA = AutoRegressive Integrated Moving Average (in this context, "integration" is the reverse of differencing)." (Source: https://otexts.com/fpp2/non-seasonal-arima.html)

The resulting mean of residuals is: [1] 22.10096

Mean of the residuals is not zero, which is suggesting that there is a bias.

The results of the checkresiduals() are:



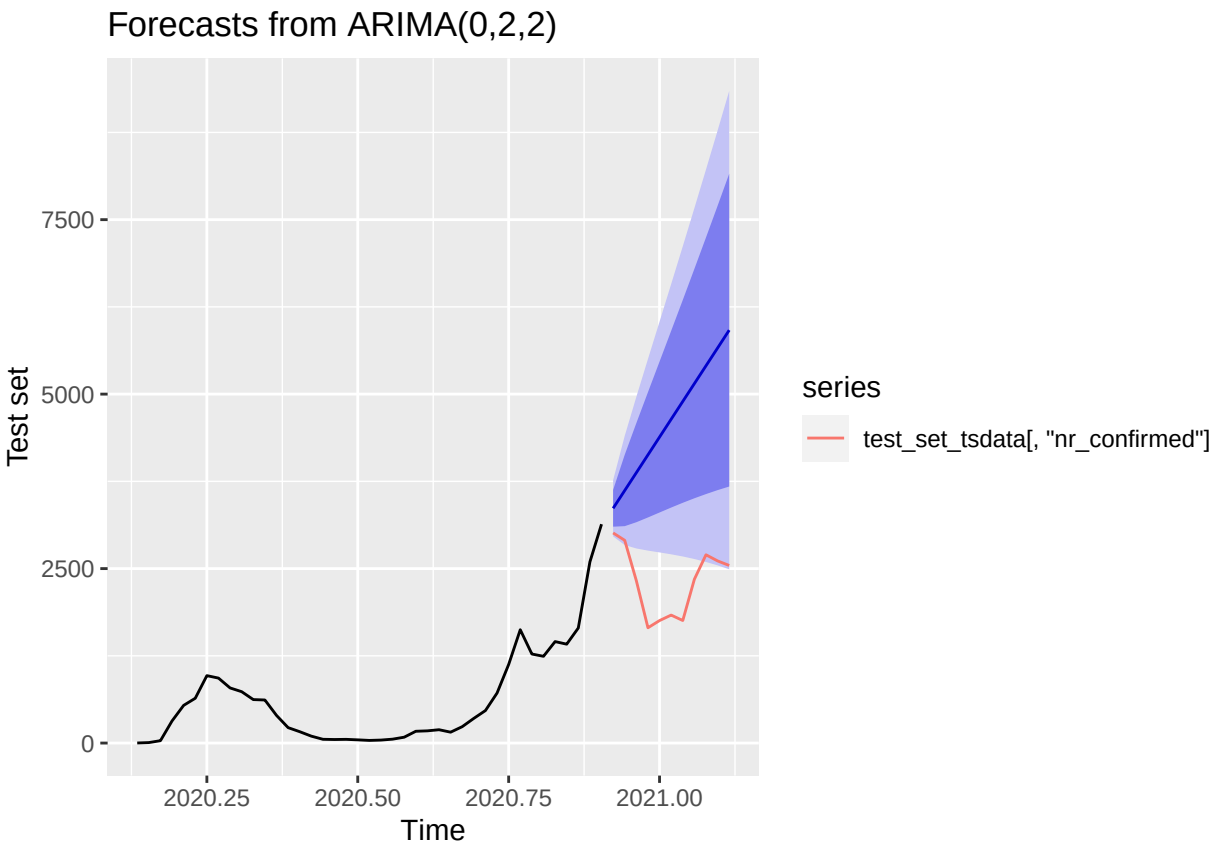
Ljung-Box test

data: Residuals from ARIMA(0,2,2) Q* = 6.9241, df = 6, p-value = 0.3279

Model df: 2. Total lags used: 8

The conclusions from checking the residuals:
* The variations in the residuals do not stay the same over time,
* ACF of residuals suggests that Arima model has captured all the elements affecting the forecast,
* Residuals do not appear to be normally distributed,
* from Ljung-Box test is that as the p-value is over 0.05. Thus, we reject the null hypothesis of the test and conclude that the data values are independent, i.e. the residuals are just white noise.

Below plot shows the visualization of the point forecast results compared to the observed values in the test set:



Conclusion: the plot confirms that the test data does not fit in the 95% prediction interval.

Again the difference between an observed value and its forecast was checked using accuracy(), where the forecast errors were calculated against the test set.

method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Simple exponential smoothing	Test set	947.4104	823.3097	41.65701	756.8346
Naive	Test set	947.4573	823.3636	41.65945	756.7302
Auto ARIMA	Test set	2517.0923	2327.6225	109.43123	1031.4785

From the results of the accuracy metrics we can see that the model performs worse than the first two models.

2.4.6 Forecasts with Multivariate VAR Method

The VAR method name comes from vector autoregression. So far all the other models I have used have been univariate point forecasts of the confirmed COVID-19 cases based on the observations of the its own time series data. Next I used the VAR method to create a multivariate time series model that uses all of the other related time series with the confirmed cases time series:

- nr of tested,
- nr of deaths,
- nr of symptoms reported,
- nr of urgently referred to treatment.

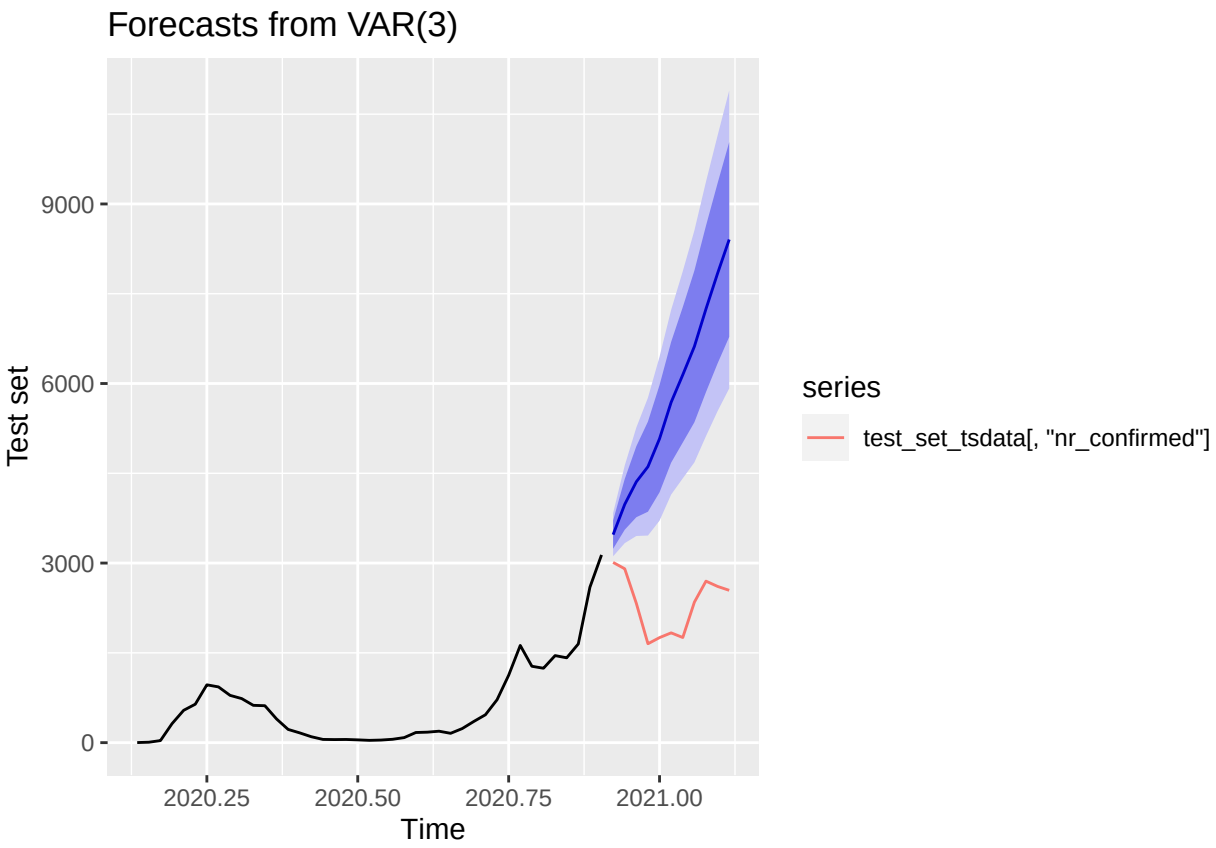
In other words I selected to use all the other time series data except vaccinations, which did not have time series data available for the whole period.

I used the VARselect() to select the number of lags p using for the different VAR(p). AIC: [1] 6 and SC (or BIC): [1] 6

The VAR() model lag was determined using the VARselect() and serial.test() was used to test that the residuals are uncorrelated according to a Portmanteau test. The Portmanteau test p-value is: Chi-squared 0.2889199

VAR(3) was chosen to fit as it passes the Portmanteau test for serial autocorrelation, since the p-value is greater than the significance level of 0.05.

Below plot shows the visualization of the point forecast results compared to the observations in the test set:



Conclusion: the plot shows that the observed test data does not fit in the 95% prediction interval.

The difference between an observed value and its forecast was checked using accuracy(), where the forecast errors were calculated against the test set. Note that the tsCV() used on the univariate methods cannot be used for VAR() as it only works on univariate time series.

method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Simple exponential smoothing	Test set	947.4104	823.3097	41.65701	756.8346
Naive	Test set	947.4573	823.3636	41.65945	756.7302
Auto ARIMA	Test set	2517.0923	2327.6225	109.43123	1031.4785
VAR	Test set	3814.6889	3455.4347	159.06295	NA

From the results of the accuracy metrics we can see that the model performs the worst of all the models.

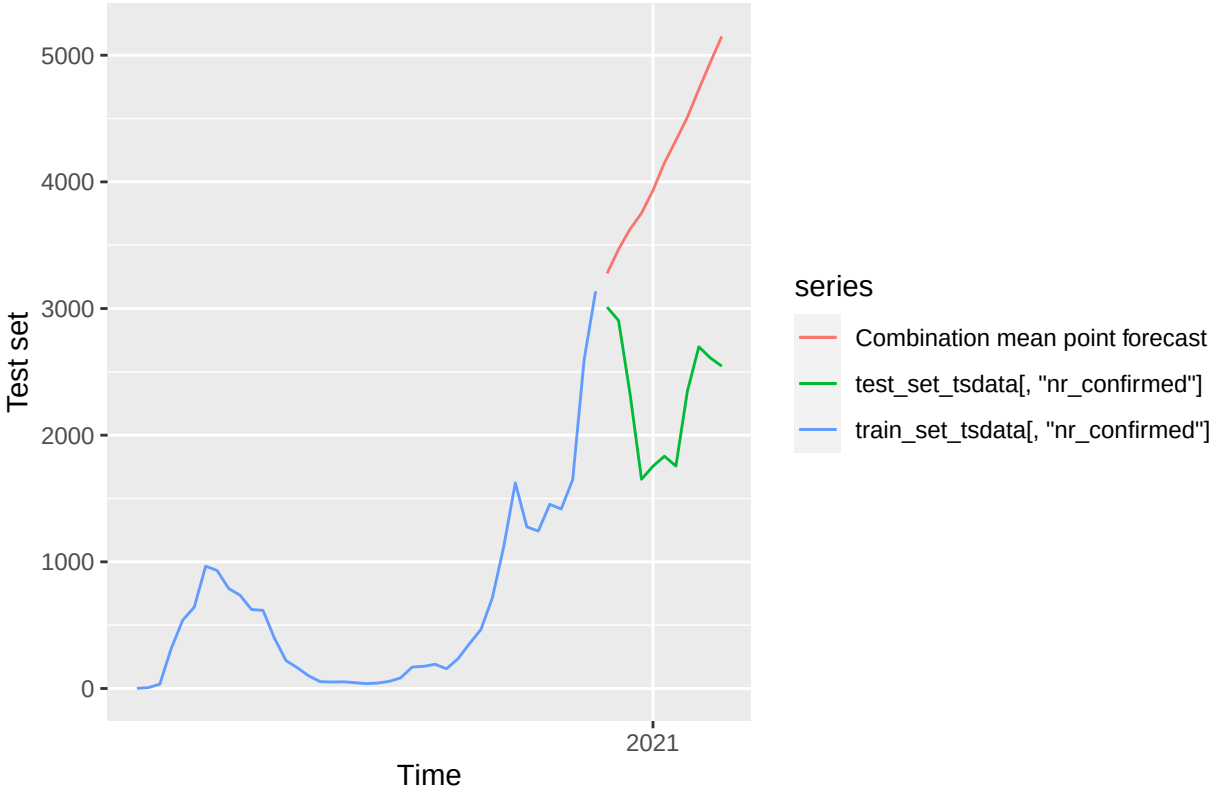
2.4.7 Forecasts with combined model

Similarly to what we learned in the Machine Learning course in relation to predictions, the combinations of point forecast models produce typically the best results. (source: https://otexts.com/fpp2/combinations.html) Therefore, the last model I developed was a combination model.

I calculated the combination model simply by taking the mean point forecasts of each model and then took the average of those means. The plot below visualizes the results of the different models.

I did try to run the checkresiduals() on the combination model, but it failed to find appropriate degrees of freedom for the model. I assume this due to the fact that the combination model even though it is a time series object, it is not a real forecast object.

Plotting the train set, test set and the combination model point forecasts alone looks like this.



The difference between an observed value and its forecast was checked using accuracy(), where the forecast errors were calculated against the test set. But like with the VAR() I was unable to produce the time series cross-validation RMSE. Note that the tsCV() used on the univariate methods cannot be used for the combined model as it only works on functions returning an object of the class forecast.

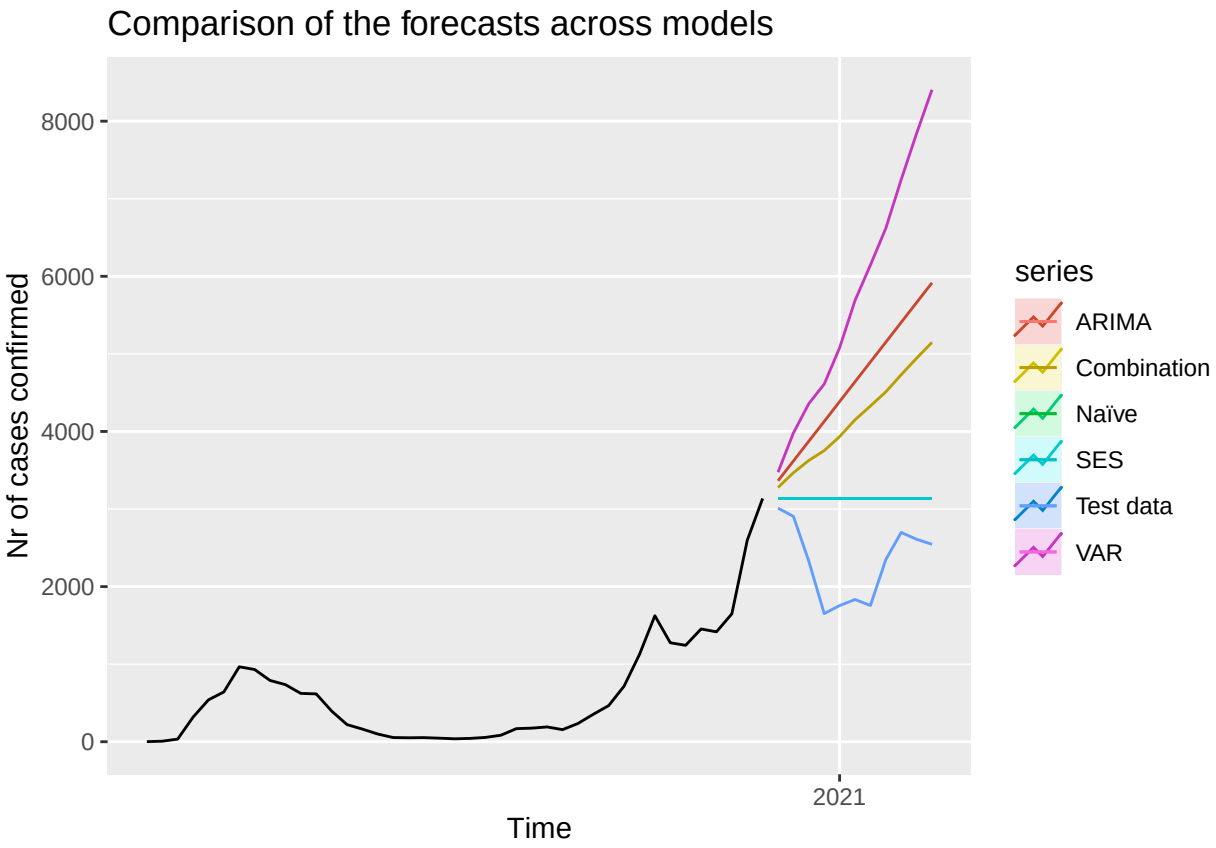
method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Simple exponential smoothing	Test set	947.4104	823.3097	41.65701	756.8346
Naive	Test set	947.4573	823.3636	41.65945	756.7302
Combination	Test set	2005.8715	1857.4326	87.95266	NA
Auto ARIMA	Test set	2517.0923	2327.6225	109.43123	1031.4785
VAR	Test set	3814.6889	3455.4347	159.06295	NA

As would be expected the accuracy of the Combination model falls between the first two models predicting straight line and the Auto ARIMA and VAR -models.

---

3. The Results

The plot below you can see that Naive and Simple exponential smoothing are practically the same straight line, however, all of the other models point forecast are showing increasing trend.



method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Simple exponential smoothing	Test set	947.4104	823.3097	41.65701	756.8346
Naive	Test set	947.4573	823.3636	41.65945	756.7302
Combination	Test set	2005.8715	1857.4326	87.95266	NA
Auto ARIMA	Test set	2517.0923	2327.6225	109.43123	1031.4785
VAR	Test set	3814.6889	3455.4347	159.06295	NA

Conclusion: The multivariate VAR performed worst on all metrics and Simple Exponential Smoothing (SES) performed best on all other than the RMSE of time series cross-validation, but the difference to the Naive model was tiny. Theory says that the time series cross-validation RMSE comparison gives the best results. None of the forecasting models did particularly well with the best two models (naive and simple exponential smoothing) having Mean absolute percentage error (MAPE) at best [1] 42 and the Mean Absolute Error (MAE) being off by about [1] 823 cases confirmed.

However, as I already know that the future observations are not a flat line, which is what both Naive and SES forecasts give, I'm going to do the validation with both Naive and Combination models.

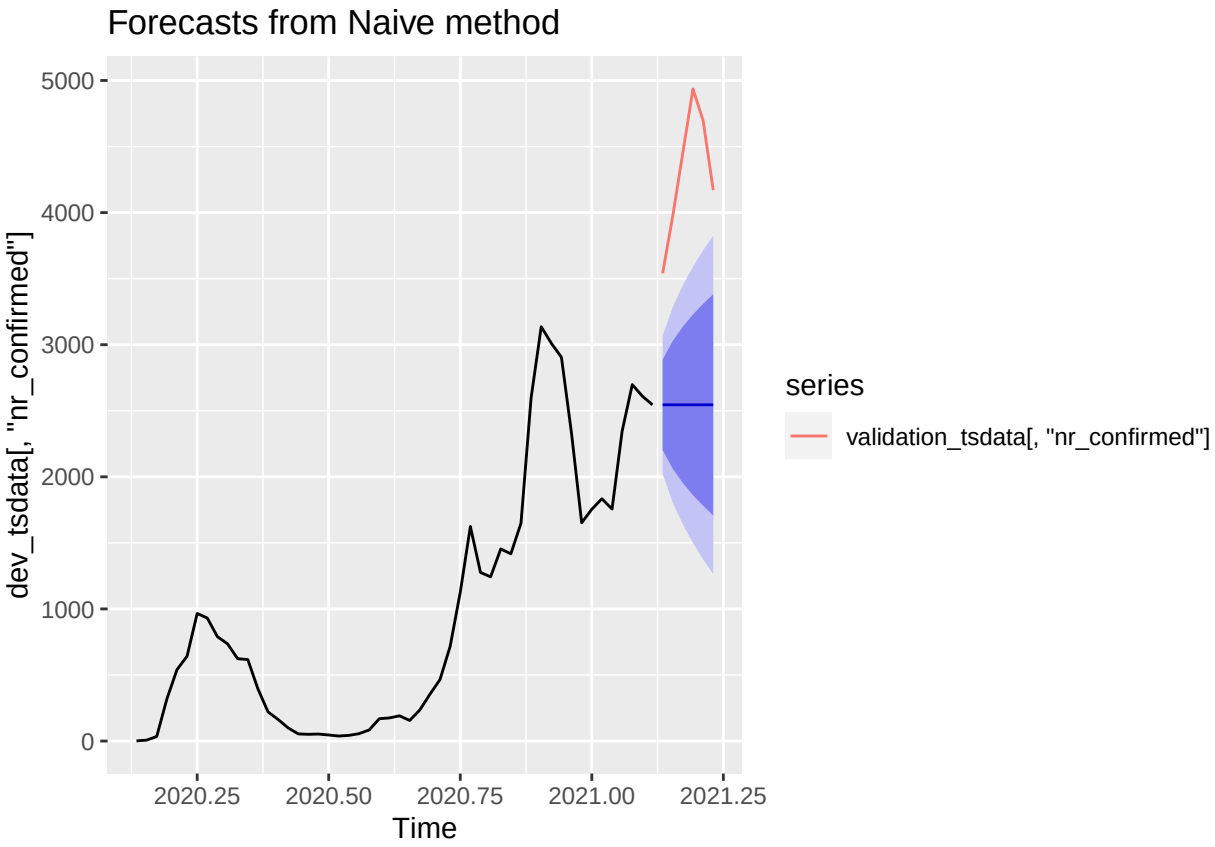
3.1 Validating the Naive model with the validation data set

As the Covid-19 data is time series data and there is a limited amount of it available, test and validation sets were created by taking specified weeks out of the original data set and used as test and validation data.

Train and test set used the data up to week 6 /2021 (90%) and the last 10% of the available data (weeks 7-12/2021) were be used as validation set.

For the validation of the forecasting model of the confirmed COVID-19 cases I used the the Naive model as it performed best on the test data.

The plot below shows the point forecast with the Naive model.



The actual observed number of confirmed Covid-19 cases is much higher than the point forecast by the Naive model.

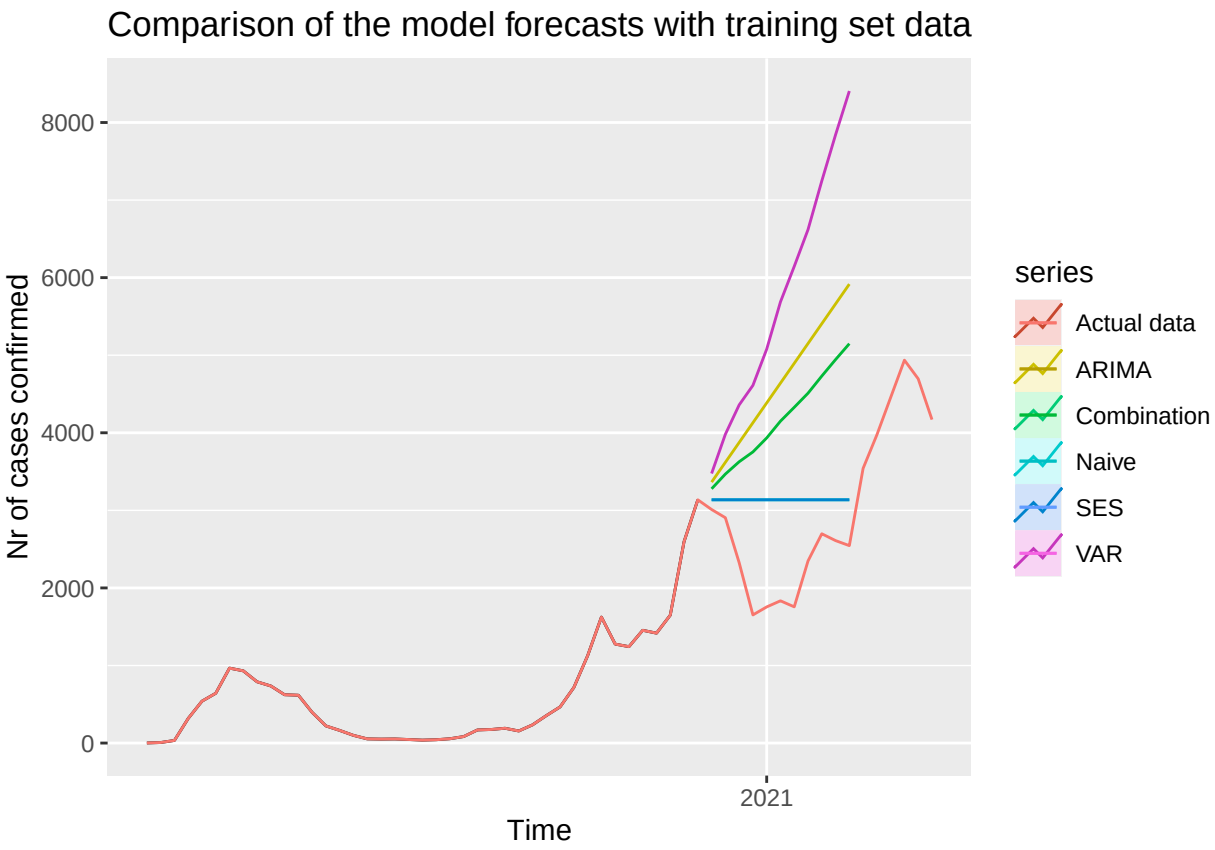
The performance metrics do not look too good either, even though the MAPE is lower than it was with the test data.

method	forecasting	RMSE	MAE	MAPE	RMSE_CV
Simple exponential smoothing	Test set	947.4104	823.3097	41.65701	756.8346
Naive	Test set	947.4573	823.3636	41.65945	756.7302
Naive	Validation set	1810.4541	1750.3333	40.02889	619.8536
Combination	Test set	2005.8715	1857.4326	87.95266	NA
Auto ARIMA	Test set	2517.0923	2327.6225	109.43123	1031.4785
VAR	Test set	3814.6889	3455.4347	159.06295	NA

The six weeks point forecast mean absolute error was [1] 1750 i.e. the forecast was that much lower than the observed values in the validation data set. In other words, the forecast did not perform well.

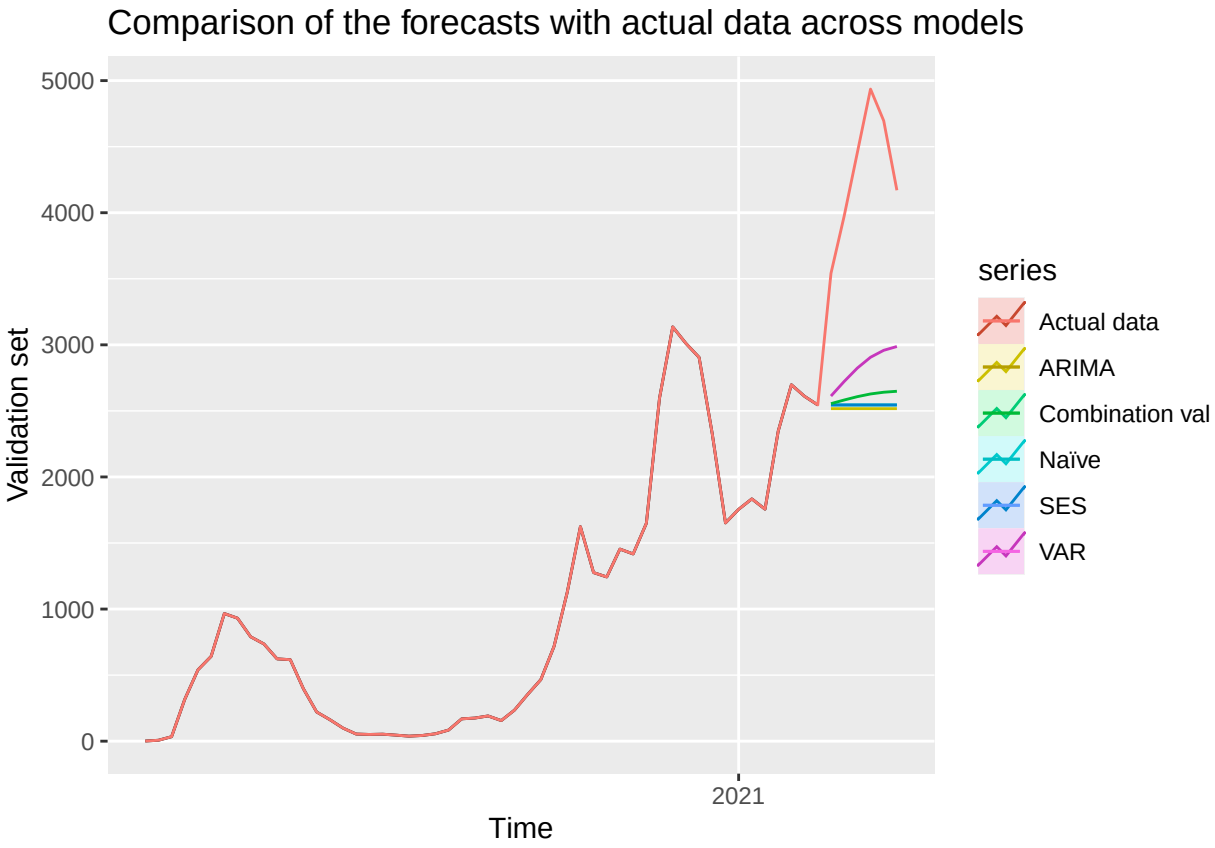
3.2 Validating the Combined model with the validation data set

Just for reference I looked at a plot with the developed models combined with the actual validation data.



On visual inspection, the original point forecast that came closest to the actual observed values was the combination model. Just to see if the combination model would have performed better than the naive model, I recalculated all of the models on the full development data set with same parameters to get the updated combination model.

Below plot displays a visual comparison of the developed models on the full development data set.



From the plot you could conclude that none of my models improved much on the mean point forecast with the addition of the test data.

---

4. The Conclusions

I started to analyze the Covid-19 data and to develop the forecast model to better understand what the data was telling in comparison to what was discussed in the Finnish press regarding the current Covid-19 epidemic status and future forecast models. I used the following methods in my model development:

- A baseline using the Naïve method,
- Simple exponential smoothing method,
- Auto ARIMA method,
- Multivariate vector autoregression (VAR) method,
- Combined model using mean point forecast of the other models.

The multivariate VAR performed worst on all metrics and Simple Exponential Smoothing (SES) performed best on all other than the RMSE of time series cross-validation, but the difference to the Naive model was tiny. As theory says that the time series cross-validation RMSE comparison gives the best results, I selected the model based on naive() method for the validation.

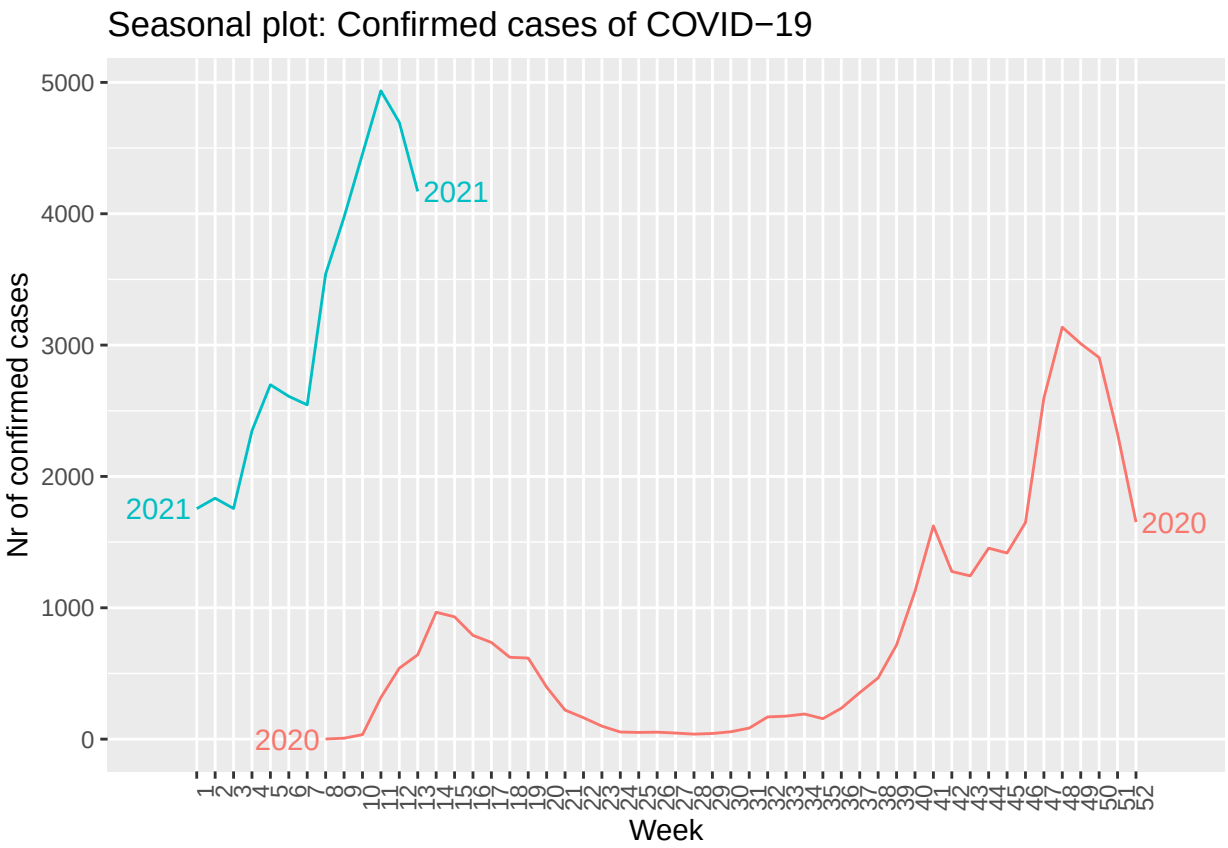
The Naive method based model on the validation data resulted in a six weeks point forecast mean absolute error of [1] 1750 i.e. the forecast was that much lower than the observed values in the validation data set. The time series cross-validation RMSE was: [1] 620 In other words, the forecast model did not perform well. I struggled to determine how important the residuals analysis was compared with the forecast accuracy. My main concern was that as the residuals for both Naive and Simple Exponential Smoothing were not indicating a good forecasting method, should I have discarded those models and only used the models with residual p-values indicating white noise? This would have meant that I should have used the ARIMA for as the final model even though it had much worse time series cross-validation RMSE than the first two models. This is something that I will need to study up in the future.

Another aspect that I did not manage to work out was the time series decomposition, i.e. if and how to deal with the trend, seasonality and cycles of the time series when developing the models using the different methods. Or if that was even possible with just 59 weeks of data.

Even though the models developed did not perform well, analyzing the available data gave me a better understanding of the changes in the trends. It also gave me a better ability to compare / contrast our situation with e.g. the situation in San Francisco, where my friend lives. The way the Covid-19 numbers are reported differs a little country by country. Here the numbers typically reported are absolute numbers, which are difficult to compare. I found the case rate seven-day average to be the easiest to comprehend and to contrast, but to get that I had to calculate it myself.

Trying to build a model to forecast the weekly confirmed cases of Covid-19 proved very difficult, which was to be expected. But after this attempt at building the forecast model I have an immense respect for the data scientists, who are building the models for the government and the health authorities. Their work is really important as it is being used as the bases for defining the actions to control the epidemic while trying to keep the economy as open as possible.

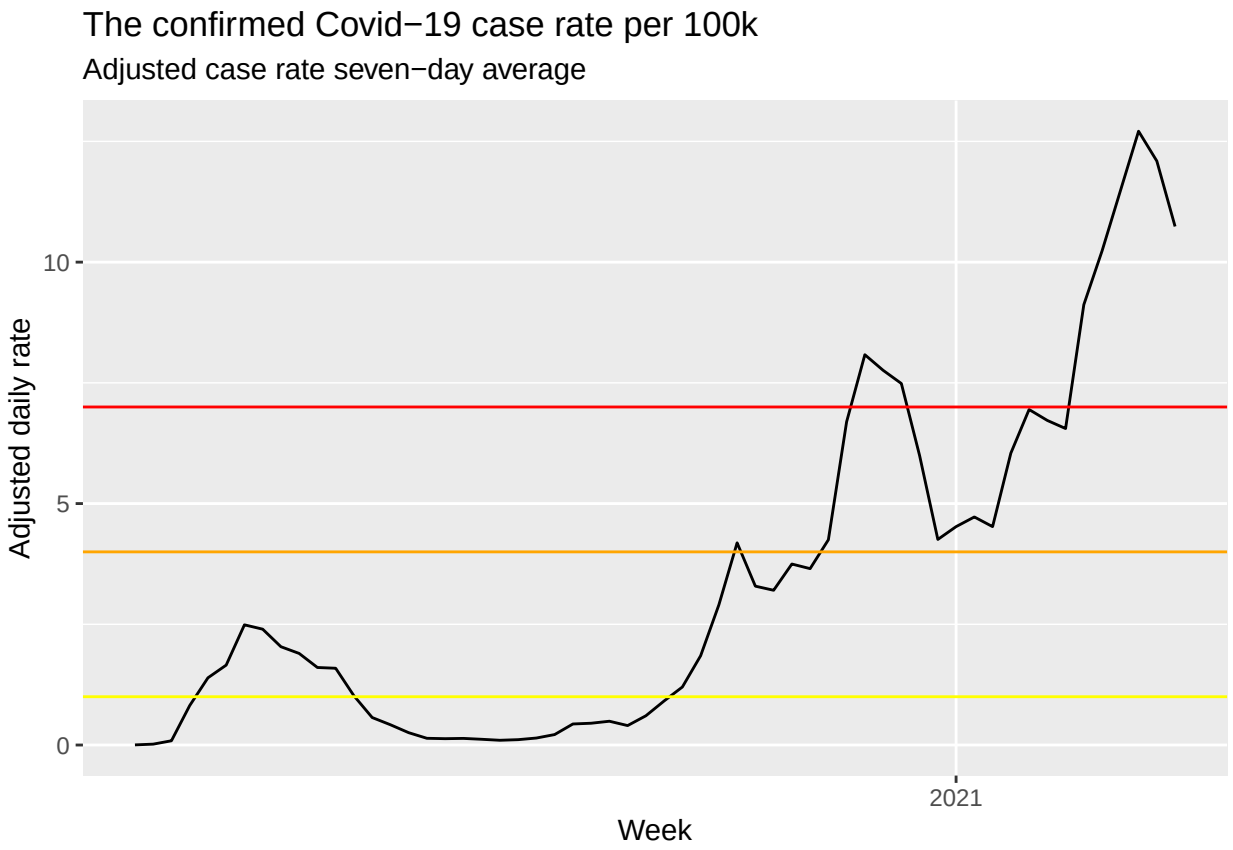
As a final action I took a look at the seasonal plot now with the full set of the data.



The plot shows that the dip in the weekly number of confirmed cases between weeks 5-7 was a temporary decrease and the trend kept on going up with about the same slope after week 7 as it had before week 5. So, you could say that the breaking point of the data into training, test and validations sets was unlucky. But that just goes to prove that forecasting the future is hard, in particular, when:

- You have no data (or don't know how to use the data) on the underlying trends or seasonality in the data.
- You do not have the data on the other variables that influence the forecasted variable.

Taking a last look at the metric of seven-day average for cases per 100 000 inhabitants.



We are still after the week 12 / 2021 at the highest level on the scale i.e. the epidemic is widespread with on average more than 10 confirmed cases per day per 100 000 inhabitants.

This project has been quite the learning experience as I accidentally selected a wrong type of data set. As I work in the field of management consulting and deal with scenario planning and forecasting in my work, I believe it was still time well spend even if it took me much longer to complete the CYO project than it should have. It also made me realize that I need to find a course that would properly teach me forecasting. I am fully aware that my models were not tuned and the methods selected were likely not using the methods best fit for the problem at hand.