Different models to forecast the spread of CoViD-19

EstebanMaureiraVenegas - emaureira.venegas@gmail.com

$08 - 01 - 2021$

Abstract

This is a project for the data science professional certificate capstone from HarvardX institution through the Edx platform and the objective is to construct a forecast the spread CoViD-19 through predictive models using updated time series worldwide data of reported cases for the Novel CoronaVirus Disease (CoViD-19) from the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) data repository(1). This report shows different methods that forecast the spread of the virus. In the first section (Sec. I) describes the dataset and variables, and summarizes the goal of the project and key steps that were performed. In the second section (Sec. II) explains the process,techniques used, data cleaning, data exploration and visualization. To model were used with R package fpp2. The third section (Sec. III) shows the results presents the modeling results and discusses the model performance. And the last section shows the conclusion that gives a brief summary of the report (Sec. IV).

Keywords: Forecast, Fpp2

1. Introduction

Forecasting is required in many situations: deciding whether to build another power generation plant in the next five years requires forecasts of future demand; scheduling staff in a call centre next week requires forecasts of call volumes; stocking an inventory requires forecasts of stock requirements(2). In this report we want to show different models to forecast CoViD-19 from my country (Chile) and also we show their respective accuracy. It used *root mean squared error* (RMSE) as a metric.The data is taken from the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) data repository (3) are useful because they can provide a forecast for CoViD-19 pandemic to effectively control the spread of this highly infectious disease in Chile. Depending on the predictions, the government officials should adapt aggressive interventions to control the growth of this rapid infectious disease the CoViD-19 pandemic. The updation of Johns Hopkins university on a daily basis is considered and the data for Chile till 23 th February 2021 is considered for this analysis and a time-series. Exploratory data analysis of was performed to predict, on a short term, confirmed cases of CoViD-19 in Chile for the last 30 days. 30 days is a reasonable amount for the short data record required for modeling and is close to 10% of the time-series. To show the precision of each model, we will use for train-set, the time-series that comprises from the beginning of the record (in this case, February 23, 2020 with the first cases in Chile) until today, subtracting the last 30 days. And we will use those last 30 days for our test-set.

2. Methods

2.1 Data

Patient data were obtained from the official website of the Johns Hopkins University Center for Systems Science and Engineering (3) that reports latest information of CoViD-19 infection in Chile. The data model development was done based on the update of February 23, 2020 with the first cases. Patient data consisted of three groups, namely registered confirmed cases, recovered cases and death cases.

```
df  %>% as_tibble() %>% dim()
```

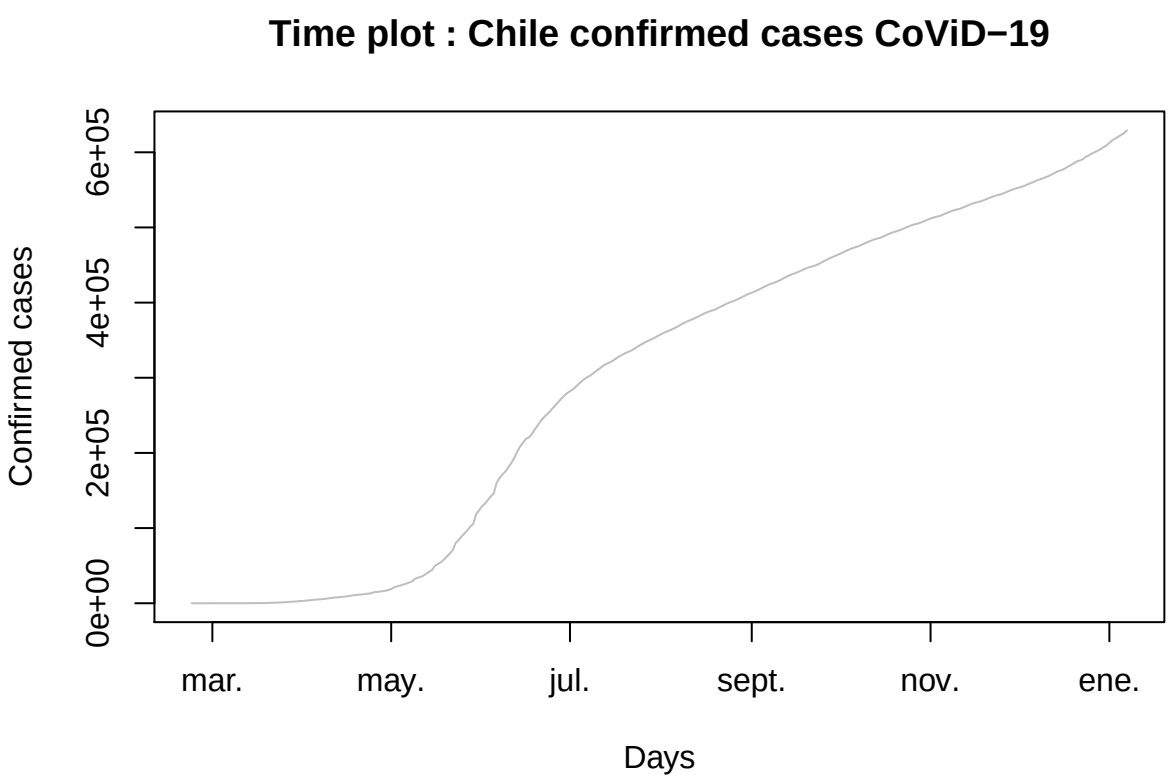
```
## [1] 320   2
```

```
df  %>% as_tibble() %>% head()
```

date	Confirmed_cases
2020-02-23	2
2020-02-24	2
2020-02-25	2
2020-02-26	2
2020-02-27	2
2020-02-28	2

In this study, we use the confirmed cases to forecast. Rather than observing entire data, we only considered observation from 23 February 2020. The next figure is a plot of total number of registered cases trend varied on daily basis.

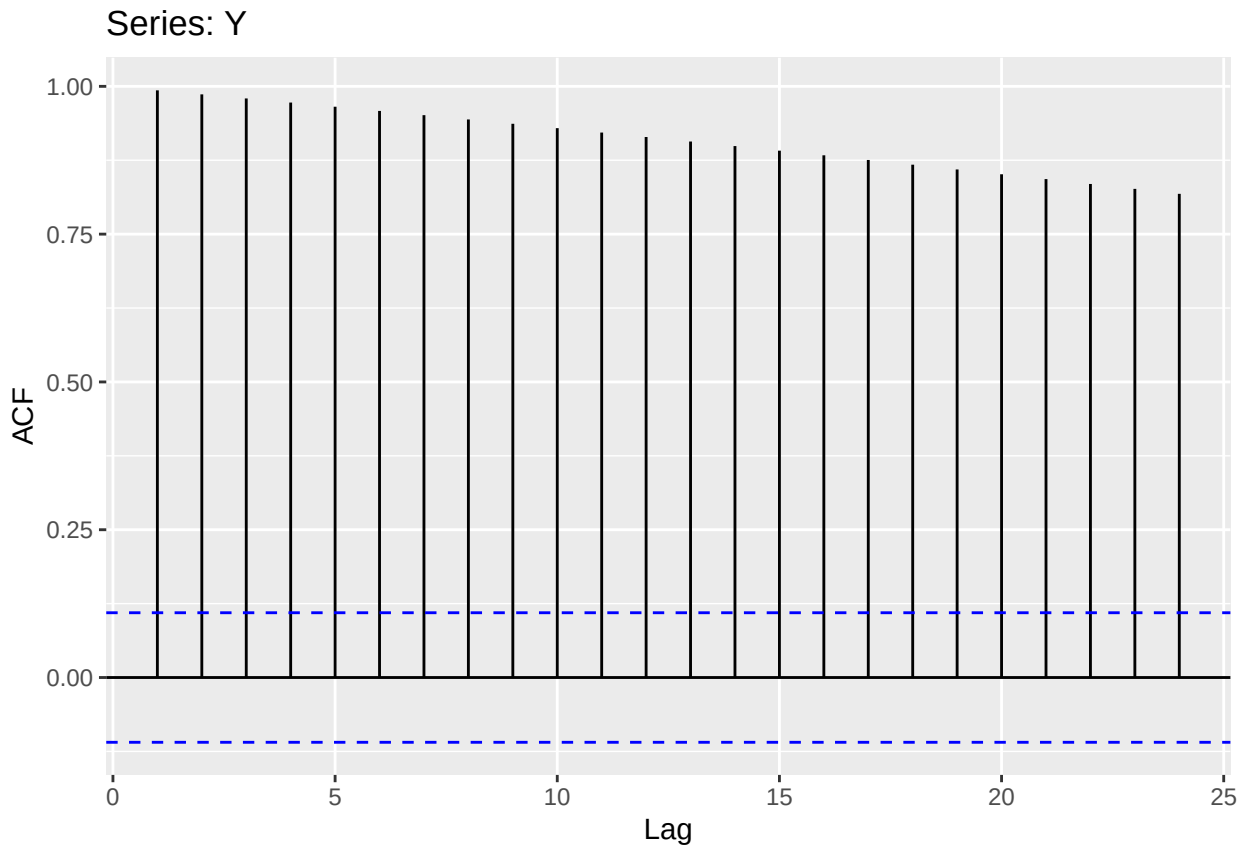
```
plot(df,type = "l", col = "grey", xlab = "Days", ylab = "Confirmed cases",
     main = "Time plot : Chile confirmed cases CoViD-19")
```



Note that the time-series presents a trend with a very clear positive slope and not stationary behavior. It is also observed that it does not have a cyclical and seasonal behavior, which can be answered by the short record of the disease.

When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So the ACF of trended time series tend to have positive values that slowly decrease as the lags increase(2).

```
ggplot2::autoplot(ggAcf(Y, lag=24)) # We can see the correlation with ACF graph.
```



Note that the slow decrease in the ACF as the lags increase is due to the trend.

2.2 Data preparation

The data is in csv format, it was not necessary to do a lot of cleaning (only the NA values and filter the date to start with the first confirmed case in Chile). To work modeling with FPP2, the time series was adapted in .TS format. The time-series starts February 23, 2020 and ends on January 08,2020. A horizon of 30 days was used for the forecast, so the last 30 days made up the test set, and the rest of the days, 290, the train set.

2.3 Modeling

The following models were used for forecasting.

2.3.2 Naïve method

This method works remarkably well for many economic and financial time series. The naïve forecasts, it is set all forecasts to be the value of the last observation (2).

$$\hat{y}_{T+h|T} = y_T.$$

Where $\hat{y}_{T+h|T}$ is a short-hand for the estimate of y_{T+h} based on the data $y_1, \dots, y_T$..

```
checkresiduals(fit_naive)
```

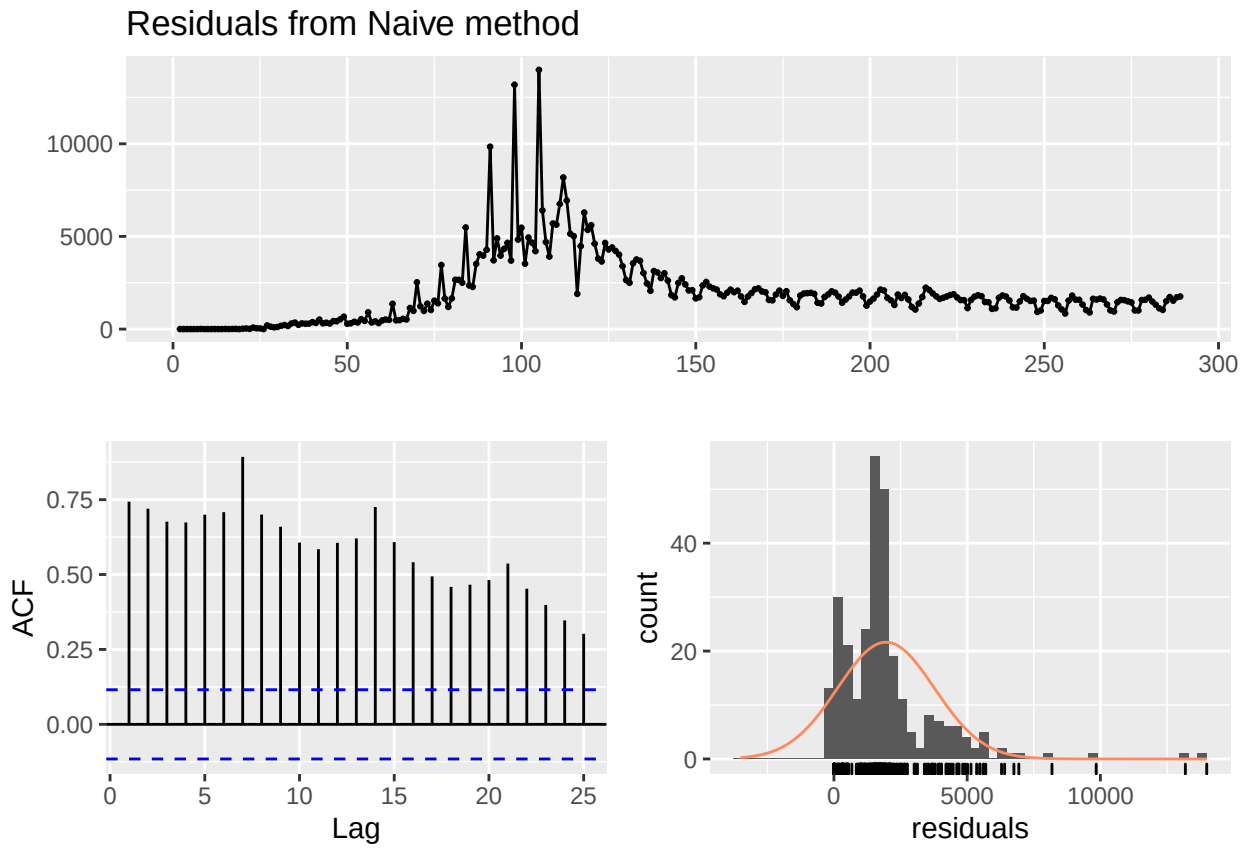


Figure 1: Naive Residual

```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 1496.7, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

Note that in the ACF exist a high correlation(due to the positive trend of the time-series) which tells us that there is information left in the residuals which should be used in computing forecasts and therefore the model is not optimal. For a good model, the residual must behave like a normal distribution, which does not occur in this case.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(fit_naive, series="naive model") +
  ggtitle("Forecasts from naive method") + xlab("Days") +
  ylab("Confirmed cases") +
  guides(colour=guide_legend(title="Forecast"))
```

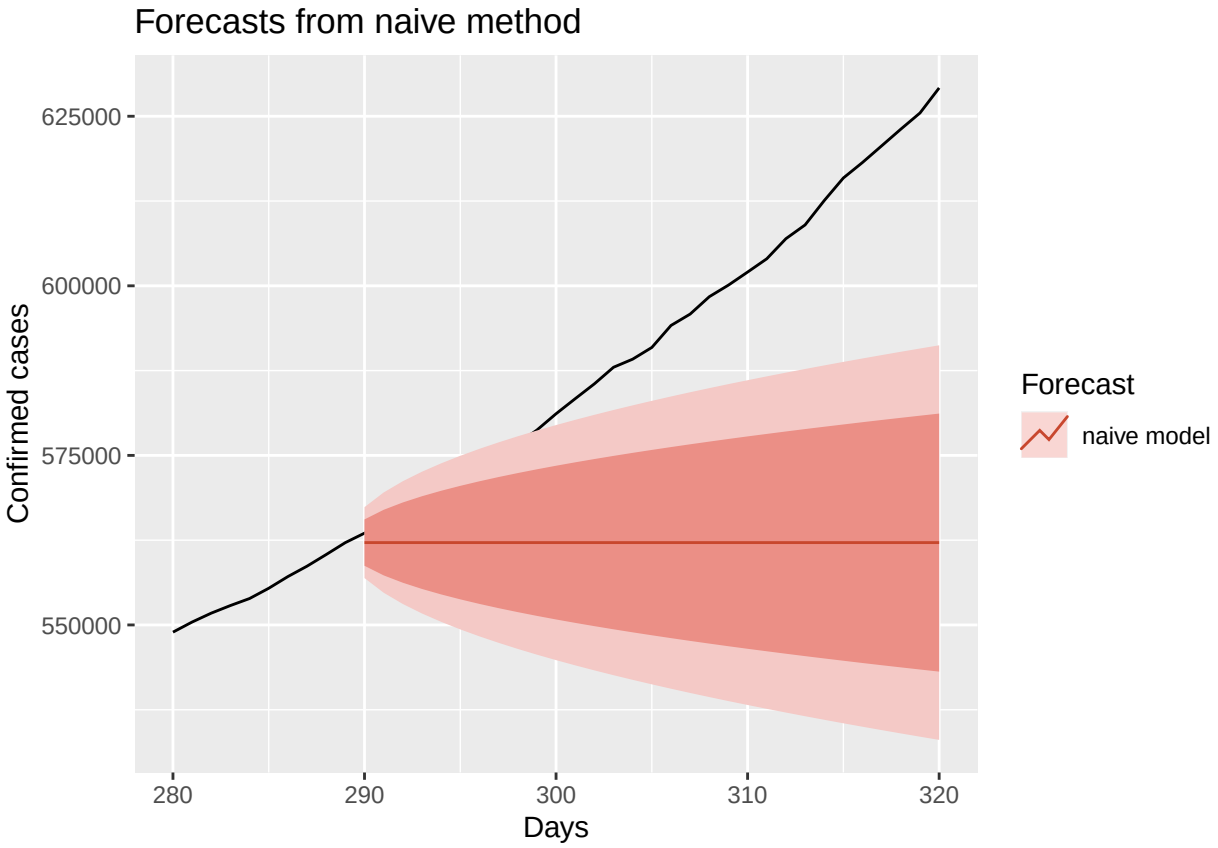


Figure 2: Naive Forecast

Predictive and confidence intervals (CI) of registered case model (Black line: actual data, orange line:30-day forecast, dark zone: 80% of CI, Light zone: 95% of CI).

2.3.3 ETS method

The ETS model is a time series univariate forecasting method; its use focuses on trend and seasonal components

$$y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1}).$$

```
checkresiduals(fit_ets)
```

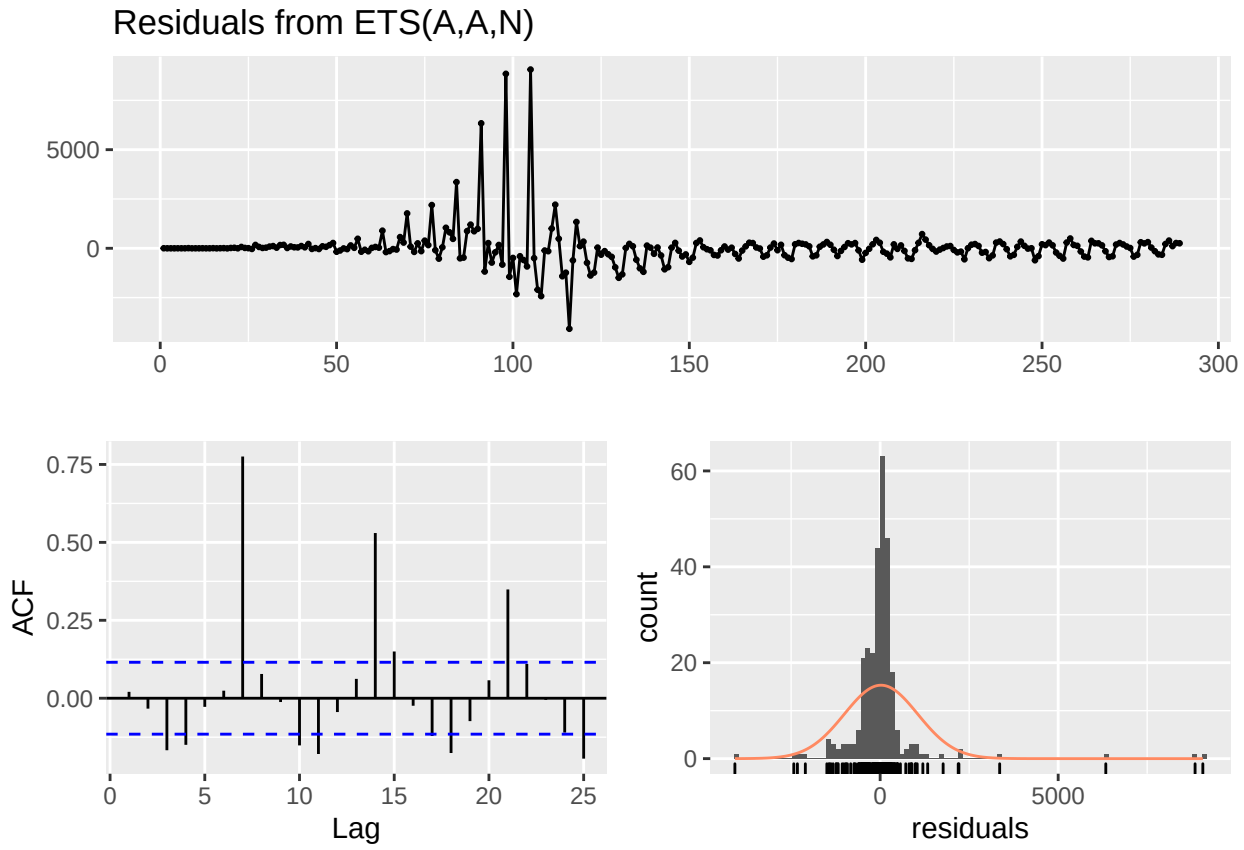


Figure 3: ETS Residual

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 203.69, df = 6, p-value < 2.2e-16
##
## Model df: 4.   Total lags used: 10
```

Note that in the ACF it could be more like a white noise as the days go by,and this is a good indication that we are not missing some information. The residual has a small behavior in normal

distribution.It isn't to bad model.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(forecast(fit_ets, h), series="ETS model") +
  ggtitle("Forecasts from ETS method") + xlab("Days") +
  ylab("Confirmed cases") +
  guides(colour=guide_legend(title="Forecast"))
```

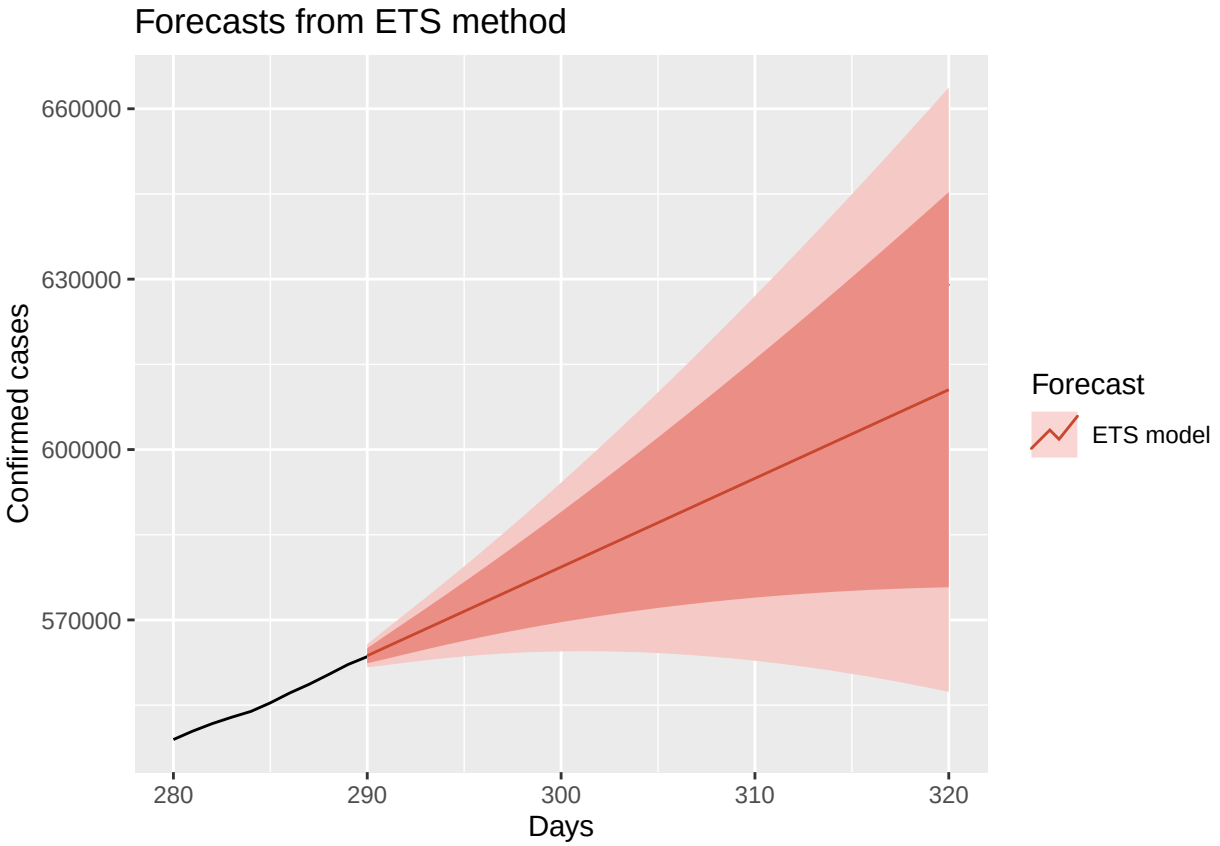


Figure 4: ETS Forecast

Predictive and confidence intervals (CI) of registered case model (Black line: actual data, orange line:30-day forecast, dark zone: 80% of CI, Light zone: 95% of CI)

2.3.1 ARIMA method

Forecasts on its previous past values and there are three distinct integers (p, d, q) that are used to parametrize ARIMA models. The three parameters account for seasonality, trend, and noise in datasets are denoted with the notation ARIMA(p, d, q). In the model, p is the auto-regressive part of the model and incorporates the effect of past values in the model. d is the integrated part of the mode and incorporates the amount of differencing to apply to the time series. The parameter q is the moving average parameter that allows to set the error of the proposed model as a linear combination of the error values observed at previous time points in the past.

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t.$$

where y'_t is the difference series (it may have been difference more than once) for Non-seasonal ARIMA model. ε_t is the error.

```
checkresiduals(fit_arima)
```

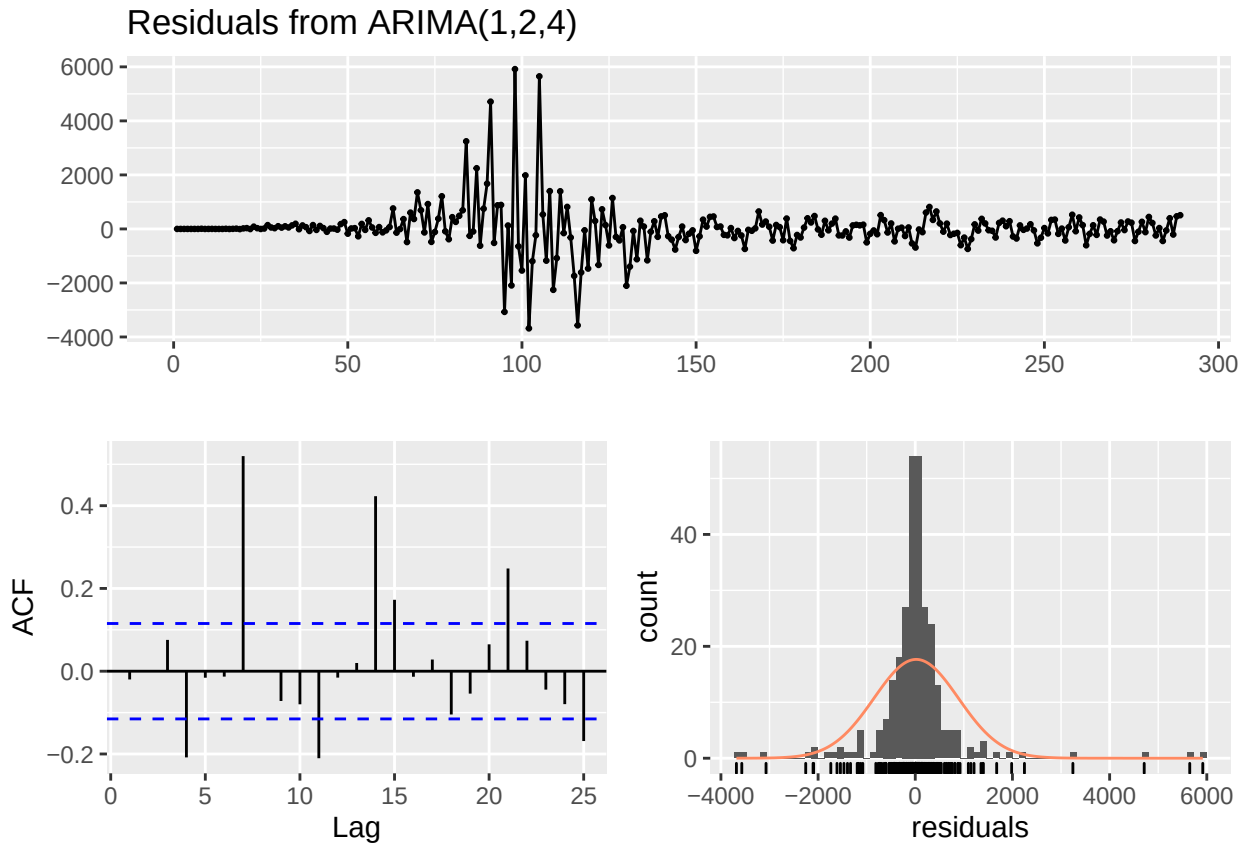


Figure 5: ARIMA Residual

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,2,4)
## Q* = 98.757, df = 5, p-value < 2.2e-16
##
## Model df: 5.   Total lags used: 10
```

Note that in the ACF it is like a white noise. The residual is like a normal distribution, centered on zero. This is a good model.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(forecast(fit_arima, h), series="ARIMA model") +
  ggtitle("Forecasts from ARIMA method") + xlab("Days") +
  ylab("Confirmed cases") +
  guides(colour=guide_legend(title="Forecast"))
```

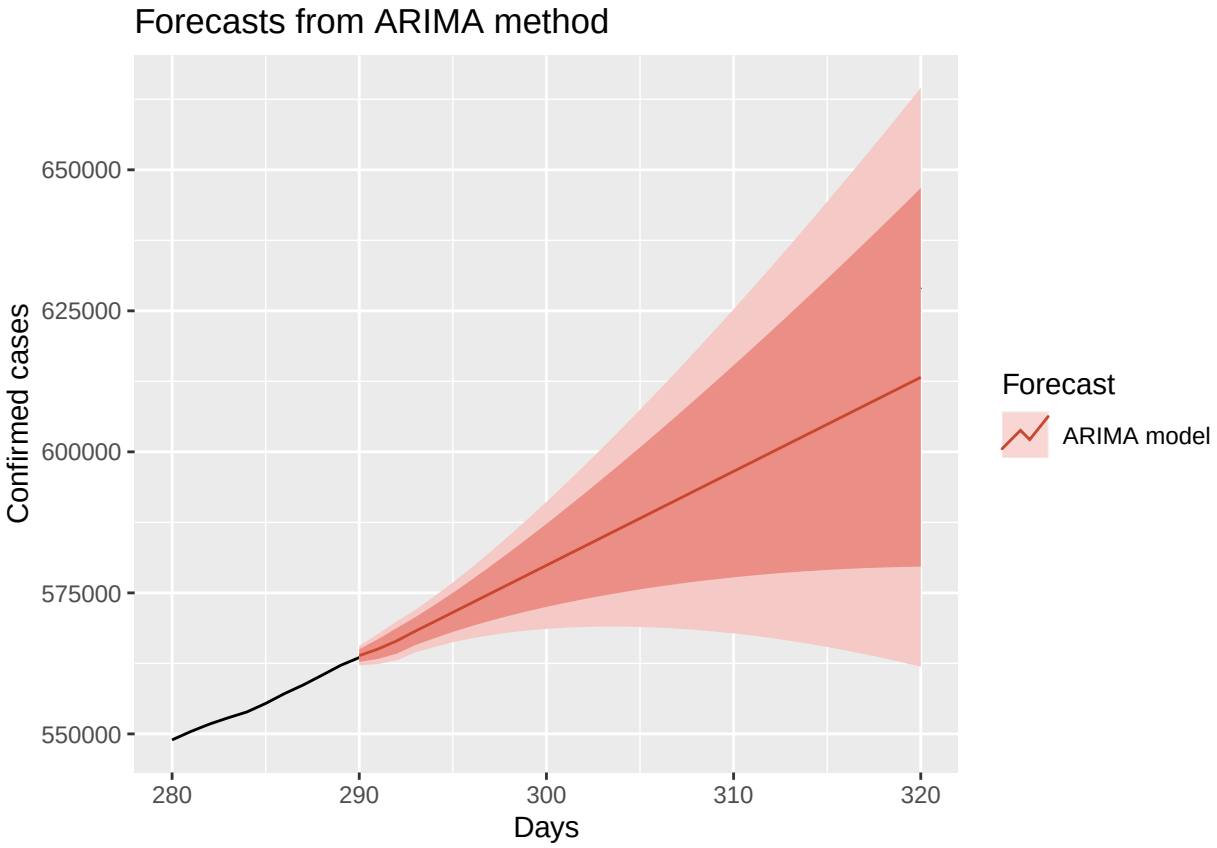


Figure 6: ARIMA Forecast

Predictive and confidence intervals (CI) of registered case model (Black line: actual data, orange line:30-day forecast, dark zone: 80% of CI, Light zone: 95% of CI)

2.3.4 Holt's linear trend method

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend)

$$\text{Forecast equation} haty_{t+h|t} = \ell_t + hb_t$$

$$\text{Level equation} ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$$

$$\text{Trend equation} b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$

where ℓ_t denotes an estimate of the level of the series at time t , b_t denotes an estimate of the trend (slope) of the series at time t , \$ \$ is the smoothing parameter for the level, \$0 1 \$, and β^* is the smoothing parameter for the trend, $0 \leq \beta^* \leq 1$.

```
checkresiduals(fit_holt)
```

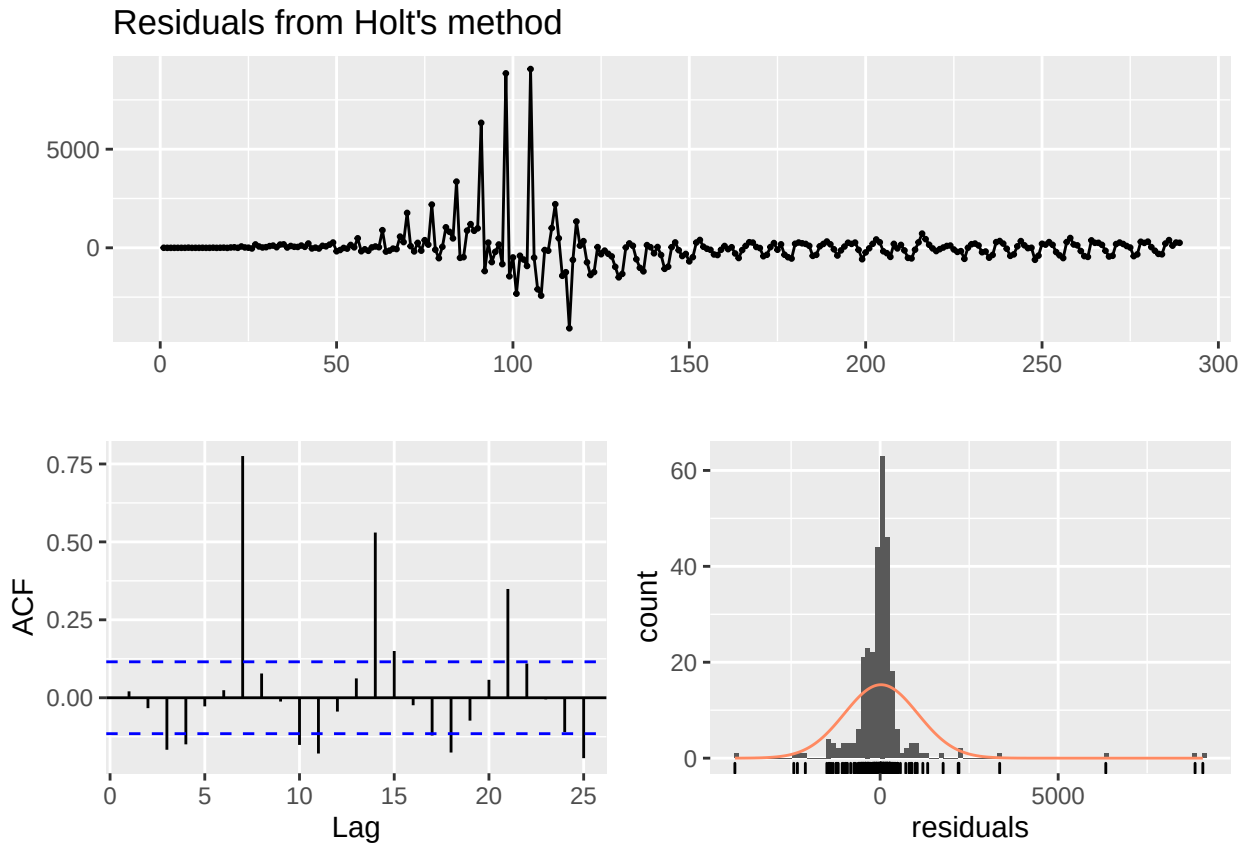


Figure 7: Holt's Residual

```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 203.69, df = 6, p-value < 2.2e-16
##
## Model df: 4.   Total lags used: 10
```

Note that in the ACF it is like a white noise. The residual is like a normal distribution, centered on zero. This is a good model.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(fit_holt, series="Holt method") +
  ggtitle("Forecasts from Holt method") + xlab("Days") +
  ylab("Confirmed cases") +
  guides(colour=guide_legend(title="Forecast"))
```

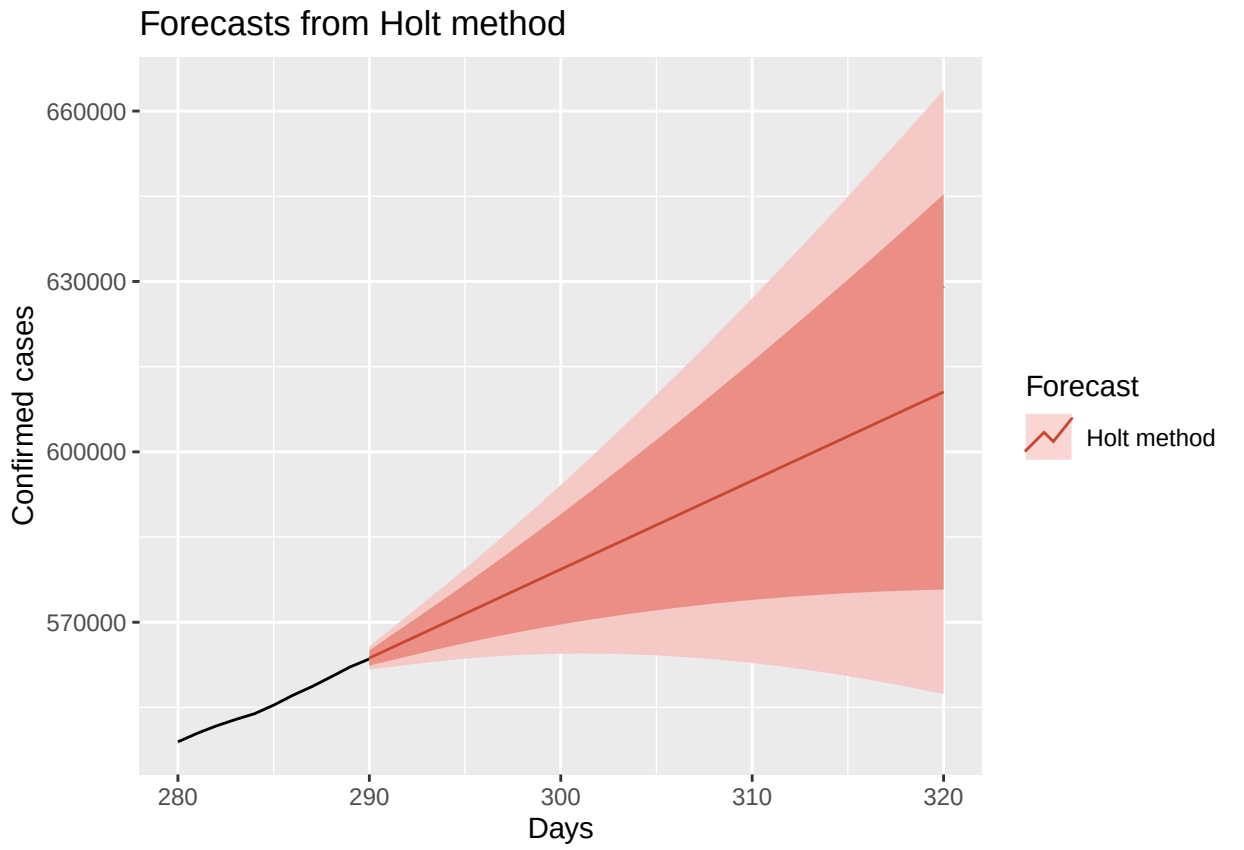


Figure 8: Holt Forecast

Predictive and confidence intervals (CI) of registered case model (Black line: actual data, orange line:30-day forecast, dark zone: 80% of CI, Light zone: 95% of CI)

2.3.5 Drift method

A variation on the naïve method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the drift) is set to be the average change seen in the historical data (2). Thus the forecast for time $T + h$ is given by:

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right).$$

```
checkresiduals(fit_rwf)
```

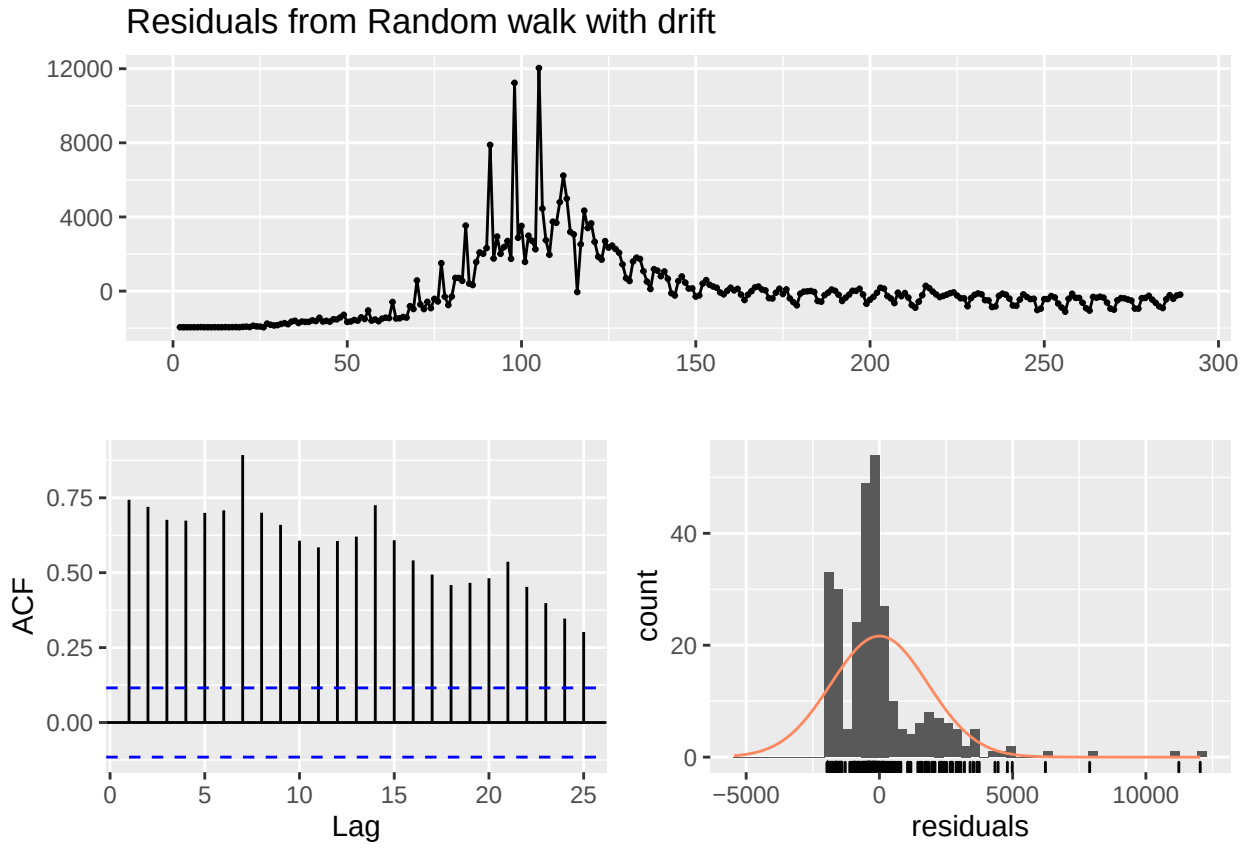


Figure 9: Drift Residual

```
##
##  Ljung-Box test
##
## data:  Residuals from Random walk with drift
## Q* = 1496.7, df = 9, p-value < 2.2e-16
##
## Model df: 1.   Total lags used: 10
```

Note that in the ACF exist a high correlation which tells us that there is information left. The residual isn't like normal distribution.It isn't to good model.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(fit_rwf, series="Drift method") +
  ggtitle("Forecasts from Drift method") + xlab("Days") +
  ylab("Confirmed cases") +
  guides(colour=guide_legend(title="Forecast"))
```

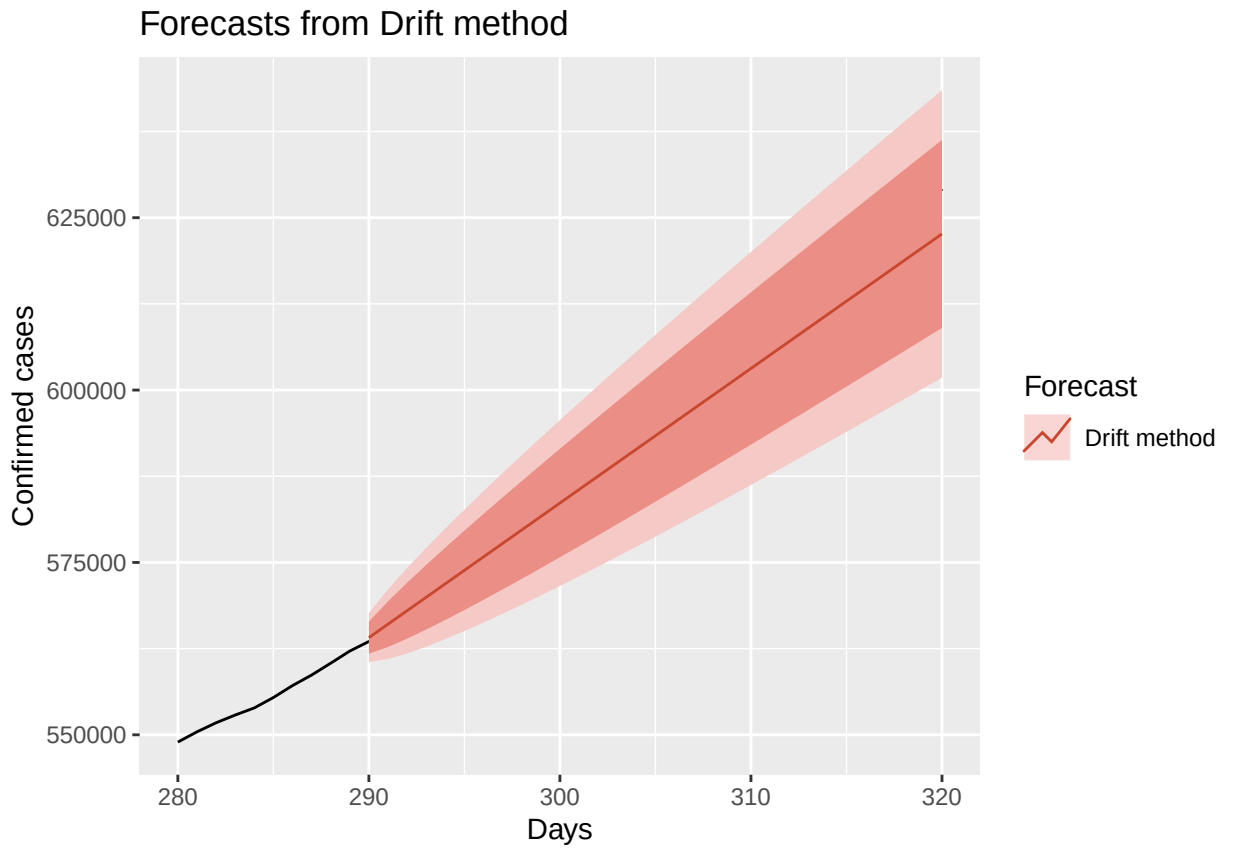


Figure 10: Drift Forecast

Predictive and confidence intervals (CI) of registered case model (Black line: actual data, orange line:30-day forecast, dark zone: 80% of CI, Light zone: 95% of CI)

2.3.6 TBATS method

An alternative approach developed by De Livera, Hyndman, & Snyder (2011) uses a combination of Fourier terms with an exponential smoothing state space model and a Box-Cox transformation, in a completely automated manner. As with any automated modeling framework, there may be cases where it gives poor results, but it can be a useful approach in some circumstances(2).

```
checkresiduals(fit_tbats)
```

```
##
##  Ljung-Box test
##
```

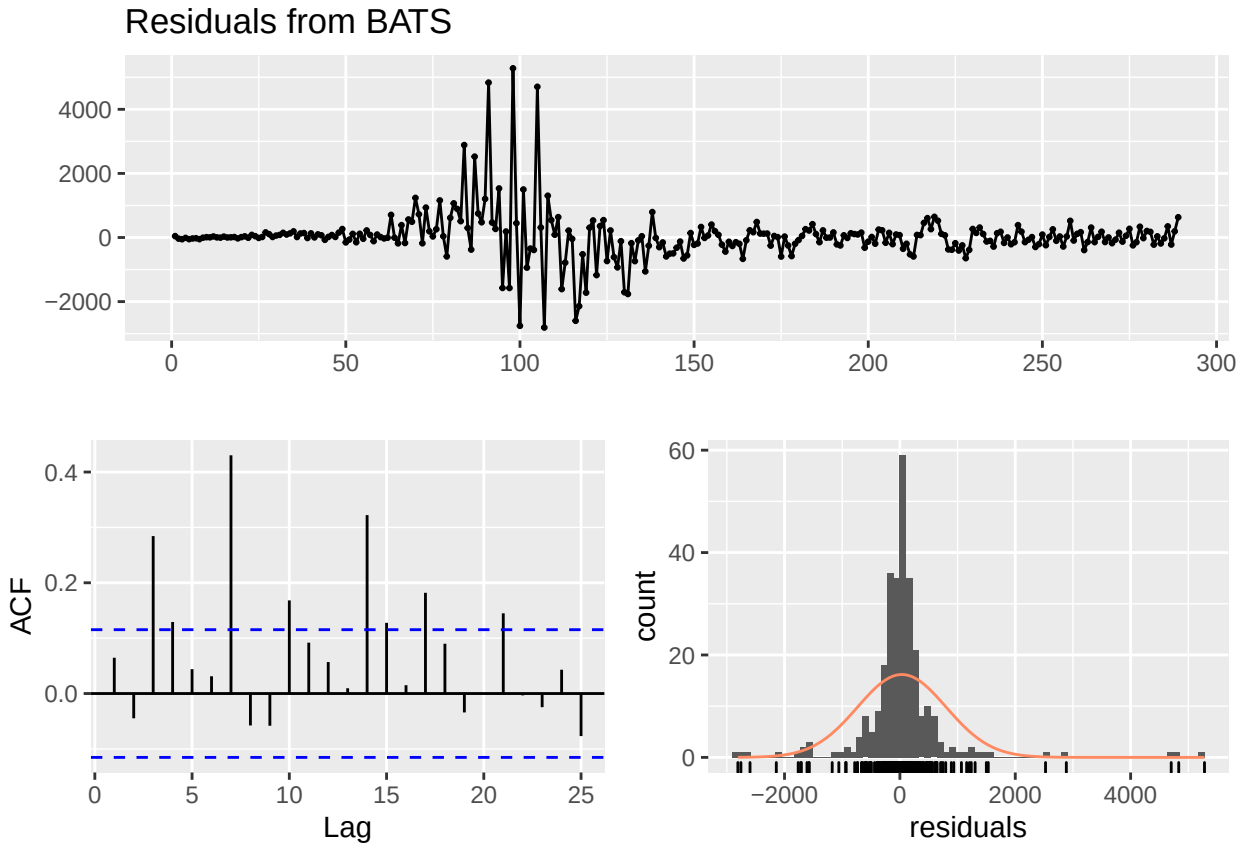


Figure 11: TBATS Residual

```
## data:  Residuals from BATS
## Q* = 157.99, df = 3, p-value < 2.2e-16
##
## Model df: 21.   Total lags used: 24
```

Note that in the ACF it is like a white noise. The residual has a normal distribution, centered on zero. This is a very good model.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(forecast(fit_tbats, h), series="TBATS method") +
  ggtitle("Forecasts from TBATS method") + xlab("Days") +
  ylab("Confirmed cases") +
  guides(colour=guide_legend(title="Forecast"))
```

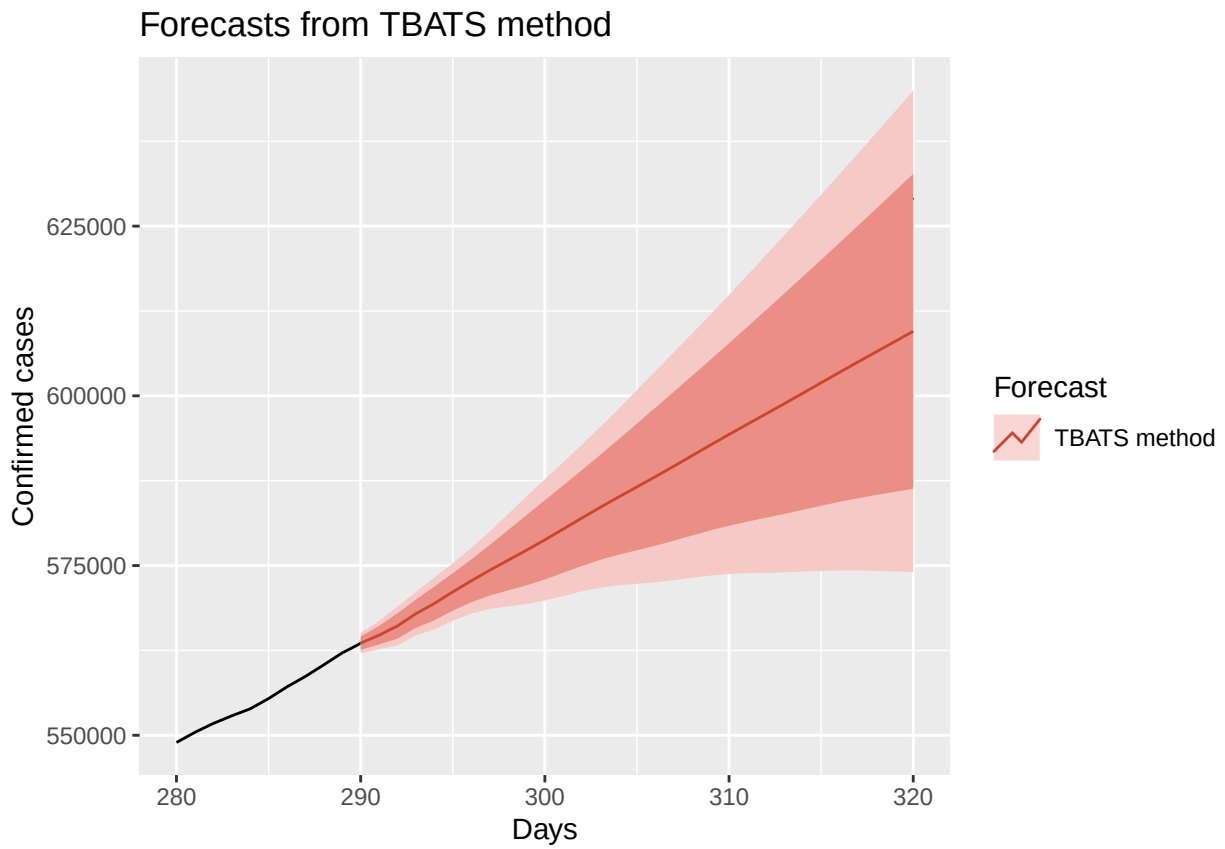


Figure 12: TBATS Forecast

Predictive and confidence intervals (CI) of registered case model (Black line: actual data, orange line:30-day forecast, dark zone: 80% of CI, Light zone: 95% of CI)

2.3.6 Combination method

An easy way to improve forecast accuracy is to use several different methods on the same time series, and to average the resulting forecasts(2). Twenty years later, Clemen (1989) wrote:

"The results have been virtually unanimous: combining multiple forecasts leads to increased forecast accuracy. In many cases one can make dramatic performance improvements by simply averaging the forecasts".

The combination, it is used the 4 best performance methods; ETS, ARIMA, Holt and TBATS methods.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(forecast(fit_ets, h), series="ETS", PI=FALSE) +
  autolayer(forecast(fit_arima, h), series="ARIMA", PI=FALSE) +
  autolayer(fit_holt, series="HOLT", PI=FALSE) +
  autolayer(fit_rwf, series="Drift", PI=FALSE) +
  autolayer(Combination, series="Combination") +
  xlab("Days") + ylab("Cases") +
  ggtitle("Forecast cases COVID19 confirmed in Chile")
```

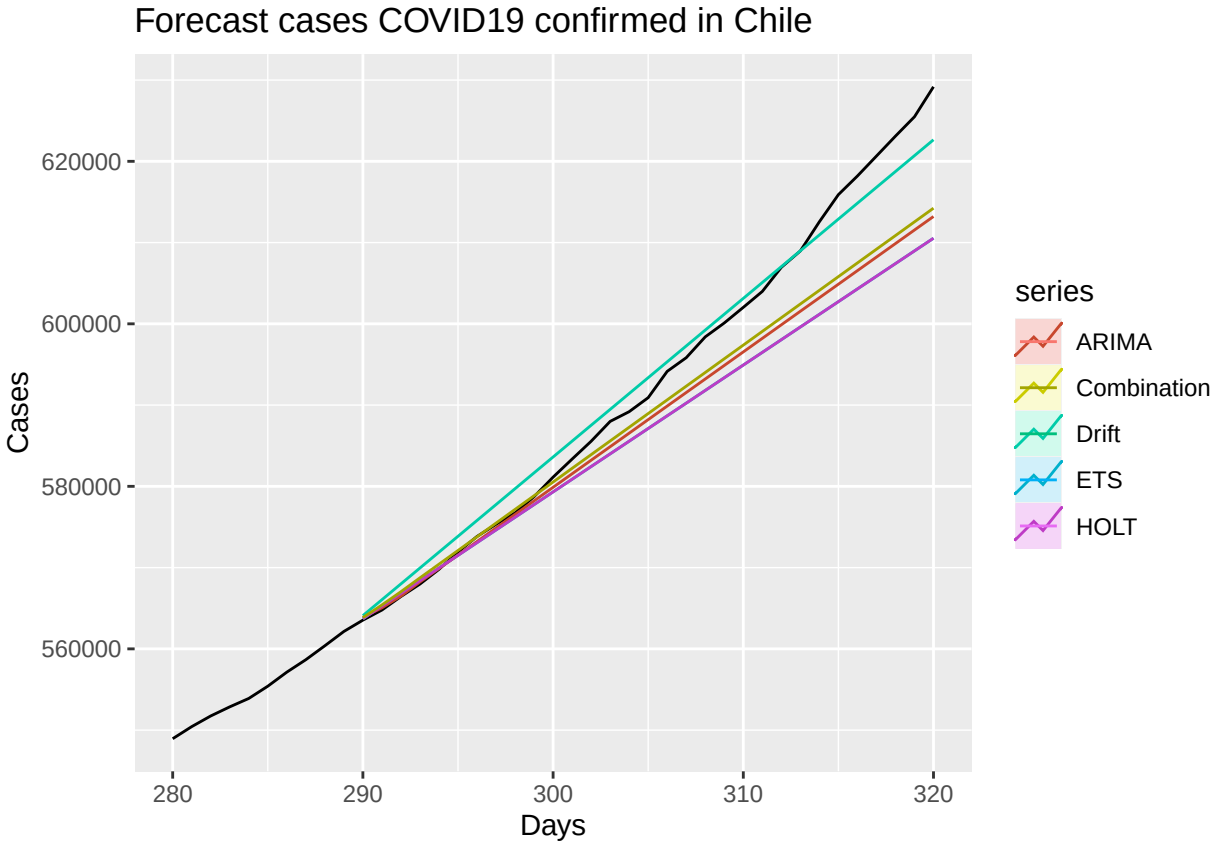


Figure 13: Combination Forecast

2.4 Evaluating Forecast Accuracy

For evaluating predictive performance, several measures are commonly used to assess the predictive accuracy of a forecasting method. In all cases, the measures are based on the test data set, which serves as a more objective basis than the training period to assess predictive accuracy. A forecast

"error" is the difference between an observed value and its forecast. Here "error" does not mean a mistake, it means the unpredictable part of an observation(2). It can be written as:

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

where the training data is given by $\{y_1, \dots, y_T\}$ and the test data is given by $\{y_{T+1}, y_{T+2}, \dots\}$. The forecast errors are on the same scale as the data. Accuracy measures that are based only on e_t are therefore scale-dependent.In this report we used root mean squared error (RMSE):

$$\text{Root mean squared error: RMSE} = \sqrt{\text{mean}(e_t^2)}$$

3. Results

The best forecast performance was shown by *Drift* method with 2550.937 and the next best RMSE accuracy is the *Combination* method 6141.679 and *ARIMA* Model 6763.630. A time series starting on February 23, 2020 and ending on January 8, 2020 is used for the forecast. A 30-day horizon was used for the forecast.

```
ggplot2::autoplot(subset(Y, start = 280)) +
  autolayer(fit_naive, series="Naive", PI=FALSE) +
  autolayer(forecast(fit_ets, h), series="ETS", PI=FALSE) +
  autolayer(forecast(fit_arima, h), series="ARIMA", PI=FALSE) +
  autolayer(fit_holt, series="HOLT", PI=FALSE) +
  autolayer(fit_rwf, series="Drift", PI=FALSE) +
  autolayer(forecast(fit_tbats, h), series="TBATS", PI=FALSE) +
  autolayer(Combination, series="Combination") +
  xlab("Days") + ylab("Cases") +
  ggtitle("Forecast summary COVID19 confirmed in Chile")
```

```
#######################################

bind_rows(method =
            c("Naive","ETS","ARIMA","HOLT","Drift","TBATS","Combination"),
                    accuracy = c(accuracy(fit_naive, Y)["Test set","RMSE"],
                    accuracy(forecast(fit_ets, h), Y)["Test set","RMSE"],
                    accuracy(forecast(fit_arima, h), Y)["Test set","RMSE"],
                    accuracy(fit_holt, Y)["Test set","RMSE"],
                    accuracy(fit_rwf, Y)["Test set","RMSE"],
                    accuracy(forecast(fit_tbats, h), Y)["Test set","RMSE"],
                    accuracy(Combination, Y)["Test set","RMSE"]))
```

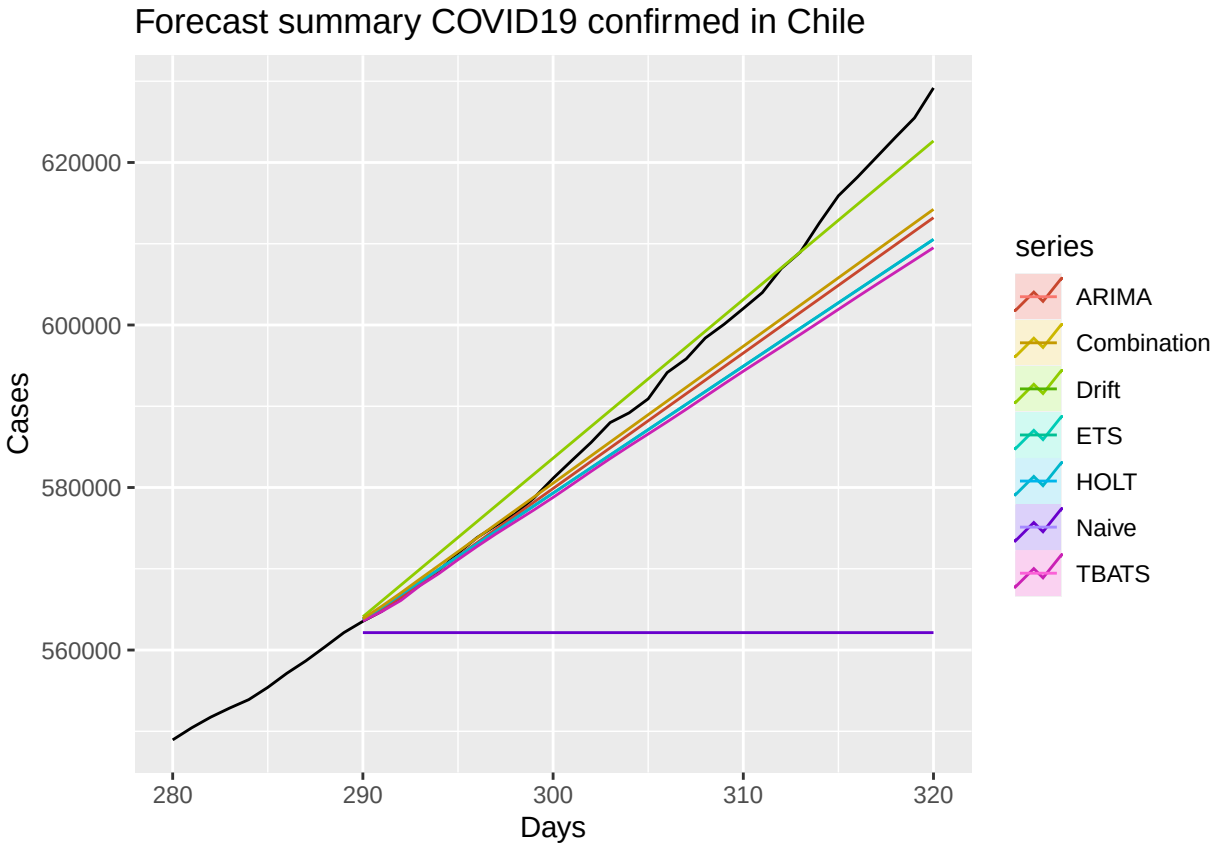


Figure 14: Forecast summary

method	accuracy
Naive	36489.274
ETS	8172.959
ARIMA	6763.630
HOLT	8172.959
Drift	2550.937
TBATS	8733.847
Combination	6141.679

4. Conclusion

It started collecting and preparing the dataset for analysis, then it explored the information seeking for insights that might help during the model building. Next, it created different models that predicts confirmed CoViD-19 cases forecasting the last 30 days in Chile (starting on February 23, 2020 and ending on January 8, 2020 is used for the forecast. A 30-day horizon).

Simple automatic forecasting fpp2 package (AUTOARIMA, Naïve, ETS, HOLT, Drift, TBATS) in 'R' was applied to conduct predictive modeling.

The best performance, in this report, was obtained by the "Drift" model, which is a variation of Naïve method. It is a simple model that adapted very well to the time-series. It should be noted that the ARIMA method performs a very flexible and adaptable modeling to different time-series.

In general, the methods made a good forecast despite the fact that it is a time-series that has not yet completed the year and that much information on its propagation behavior is still missing.

It would be very interesting for it to behave seasonally and / or cyclically, since this would give us more information with which better results could be obtained with the appropriate models.

5. References

1. covid19.analytics
2. Forecasting: Principles and Practice
3. Johns Hopkins University Center for Systems Science and Engineering