APM assignment 2

Forecasting COVID-19 Hospitalisations

Valérie BLANCH

2614867B

1. INTRODUCTION

The data set analysed in this project has been downloaded from the website https://coronavirus.data.gov.uk/details/healthcare. It contains the number of hospitalisations in the UK, per day, due to COVID-19, from 23/03/2020 to 07/07/2021.

The dataset has been tested for missing values, there were none.

```
> anyNA(covid)
[1] FALSE
```

Figure 1: NAs testing output

Originally, it consisted of 6 variables for 472 observations, but it has been converted into a time series from the single variable *newAdmissions*, since the objective of this analysis is to forecast the last two weeks of the daily admissions.

```
> dim(covid)
[1] 472   6

> covid[c(1,472),]
    areaType       areaName  areaCode       date newAdmissions cumAdmissions
1   overview United Kingdom K02000001 2021-07-07           564        476736
472 overview United Kingdom K02000001 2020-03-23          1273          4871
```

Figure 2: Raw data set structure

The resulting time series has been separated into a training set and a test set, with the following structures:

- The training set contains 457 daily observations ranging from 23/03/2020 to 23/06/2021 (83rd day of 2020 and 174th day of 2021).

```
> length(train.ts)
[1] 457

> start(train.ts)
[1] 2020   83

> end(train.ts)
[1] 2021  174
```

Figure 3: Training set structure

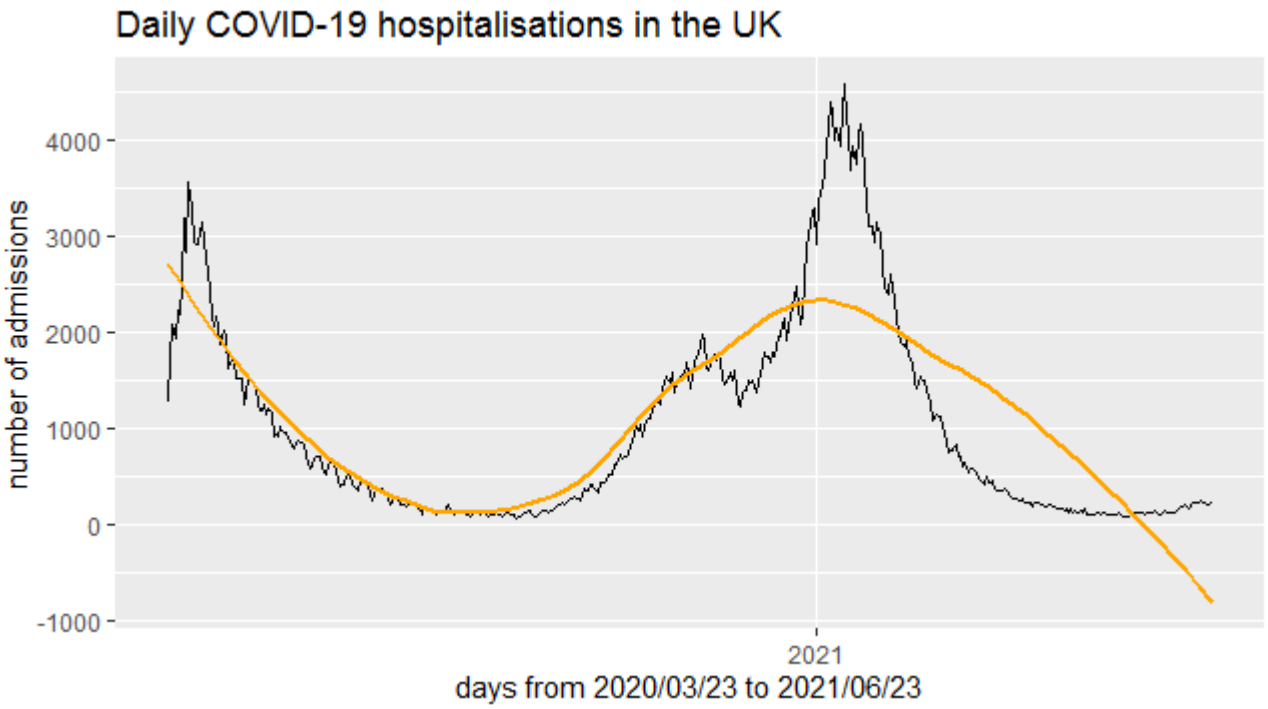
- The test set contains the last 14 observations of the original data set, from 24/06/2021 to 07/07/2021.

```
> length(test.ts)
[1] 14

> start(test.ts)
[1] 2021  175

> end(test.ts)
[1] 2021  188
```

Figure 4: Test set structure

2. EXPLORATORY ANALYSIS

An exploratory analysis has been conducted on the training set by first plotting the time series and a trend line (loess method).



- Trend: the plot shows no sign of an obvious long-term linear trend.
- Seasonality: more hindsight is necessary to confirm whether the COVID epidemic is seasonal, if it will alternate between surges of contamination and periods of stability in the long term. The decompose() function from the *fpp2* package returned an error, because it would not recognise the two peaks that can be seen on the plot as a significant pattern.
- Unexplained variation: the time series is extremely heteroscedastic and needs to be stationarized to ensure stable models and accurate forecasts.

3. RESULTS

ARIMA model - Stationarity :

To fit an ARIMA model to the data, three hyperparameters must be determined. The first one is the number of differences needed to be applied to the series to ensure stationarity.
First, the stationarity has been formally confirmed using the ADF t-test, and the following output indicates that indeed the series is not stationary, with a p-value of 0.15 under the null hypothesis that the series is not stationary.

```
        Augmented Dickey-Fuller Test

data:  train.ts
Dickey-Fuller = -3.0003, Lag order = 7, p-value = 0.1548
alternative hypothesis: stationary
```

Figure 6: ADF test output

The function ndiffs(), used to compute the number of differences needed to make the series stationary, has returned the following value:

```
> ndiffs(train.ts)
[1] 1
```

Figure 7: ndiffs output

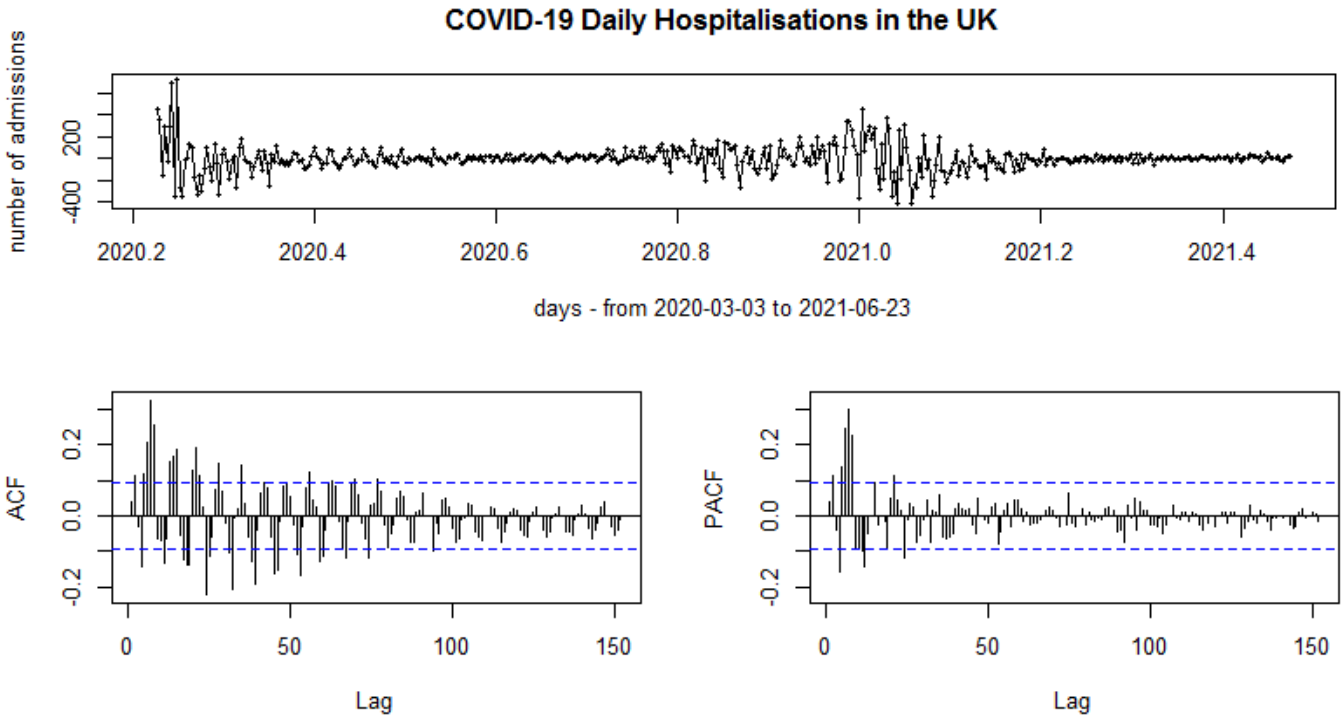
Therefore, d=1. After applying the difference to the time series, the ADF test has been re-run and gave the following output, confirming the series is stationary when d=1.

```
> train.diff <- diff(train.ts)

> adf.test(train.diff)

        Augmented Dickey-Fuller Test

data:  train.diff
Dickey-Fuller = -4.4755, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Figure 8: ADF test output after stationarization

ARIMA model - Autoregression and Moving Average:

The ACF and PACF plots have been drawn to determine the number of lags and the moving average necessary to fit the ARIMA model:



The upper graph shows the data after stationarization. On the left is the plot of the autocorrelation function: the first lag is below the significance level, therefore p=1. On the right is the plot of the partial autocorrelation function. Here again the first lag is below the significance level, therefore q=1.

ARIMA(1,1,1) model – Results:

- Modelling: to fit the model, the arima() function has been used. The ARIMA(1,1,1) model scored an RMSE of 119.72 during the training process.

```
> summary(model.arima)

Call:
arima(x = train.ts, order = c(1, 1, 1))

Coefficients:
          ar1      ma1
      0.9386  -0.8776
s.e.  0.0328   0.0413

sigma^2 estimated as 14365:  log likelihood = -2829.64,  aic = 5665.28

Training set error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE
Training set -1.485228 119.7222 71.47255 -0.5352594 9.286379 0.9788114
                   ACF1
Training set -0.0483846
```

Figure 10: ARIMA model summary output

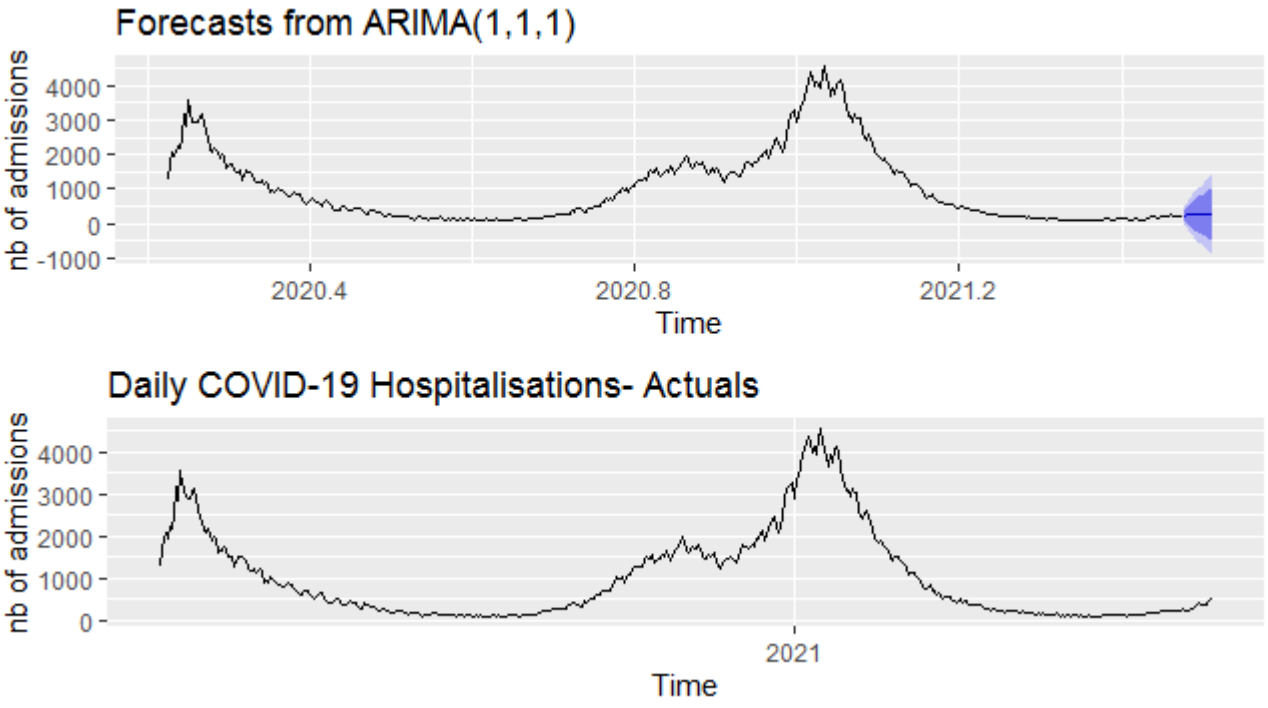
- Forecasting: predictions have been computed using the forecast() function. Below are printed their confidence intervals:

```
> forecast.arima
          Point Forecast      Lo 80     Hi 80       Lo 95     Hi 95
2021.4767       240.1121   86.51386  393.7104    5.203836  475.0204
2021.4795       242.0946   18.14854  466.0406 -100.401315  584.5904
2021.4822       243.9552  -38.27829  526.1887 -187.683683  675.5941
2021.4849       245.7016  -89.05830  580.4615 -266.269477  757.6727
2021.4877       247.3407 -136.48575  631.1672 -339.671215  834.3527
2021.4904       248.8792 -181.66199  679.4203 -409.576721  907.3351
2021.4932       250.3231 -225.19577  725.8420 -476.920279  977.5666
2021.4959       251.6784 -267.45428  770.8111 -542.266523 1045.6234
2021.4986       252.9505 -308.67268  814.5736 -605.978005 1111.8789
2021.5014       254.1444 -349.00846  857.2972 -668.298279 1176.5870
2021.5041       255.2650 -388.57086  899.1008 -729.396955 1239.9269
2021.5068       256.3167 -427.43794  940.0714 -789.395797 1302.0293
2021.5096       257.3039 -465.66697  980.2748 -848.384633 1362.9924
2021.5123       258.2304 -503.30104 1019.7619 -906.431446 1422.8923
```

Figure 11: ARIMA model forecasts & CIs

The confidence intervals are significantly large. For instance, the first prediction corresponding to the 24/06/2021 ranges from 5.20 to 475 beds (95% confidence interval).

- Accuracy of the forecasts:



The two plots above compare the predictions against the actual number of hospitalisations recorded in the test set: the model could not predict the slight increase of the last two weeks and instead produced a straight line.

```
> accuracy(forecast.arima, test.ts)
                    ME     RMSE       MAE         MPE      MAPE       MASE
Training set  -1.485228 119.7222  71.47255 -0.5352594  9.286379 0.06229371
Test set     110.550299 144.1743 110.55030 26.0960436 26.096044 0.09635291
                   ACF1 Theil's U
Training set -0.0483846        NA
Test set      0.7231668  3.111569
```

Figure 12: accuracy of ARIMA(1,1,1)

The RMSE of the forecasts is 144.17, which means that the model was able to forecast the admissions within an average error of 144 beds.

For the sake of completeness, the auto.arima() function has been used as well, to determine if ARIMA(1,1,1) was optimal. The algorithm has instead determined that ARIMA(2,1,2) was a better option, based on AIC.

```
> summary(model.auto)
Series: train.ts
ARIMA(2,1,2)

Coefficients:
         ar1      ar2      ma1     ma2
      0.5410  -0.7288  -0.4764  0.9055
s.e.  0.0549   0.0701   0.0391  0.0464

sigma^2 estimated as 13552:  log likelihood=-2814.7
AIC=5639.4   AICc=5639.53   BIC=5660.01

Training set error measures:
                    ME     RMSE      MAE         MPE      MAPE       MASE
Training set -1.868219 115.7728 70.4536 -0.9326802 9.551843 0.06140562
                   ACF1
Training set -0.04704417
```

Figure 13: ARIMA(2,1,2) summary

Though the RMSE of ARIMA(2,1,2) is lower than ARIMA(1,1,1) on the training set, it is slightly higher on the test set, as shown below:

```
> accuracy(forecast.auto, test.ts)
                     ME     RMSE      MAE         MPE      MAPE        MASE
Training set  -1.868219 115.7728  70.4536 -0.9326802  9.551843 0.06140562
Test set     112.324480 148.7777 113.7024 26.2227389 26.776521 0.09910023
                   ACF1 Theil's U
Training set -0.04704417        NA
Test set      0.73342901  3.209657
```

Figure 14: ARIMA(2,1,2) forecasts accuracy

Holt's model:

Alternatively, exponential smoothing can be used on this data set to forecast daily COVID-19 hospitalisations in the UK. Unlike the ARIMA process, exponential smoothing does not use moving average, but adds weights to past values according to their age: the heavier weights are added on the most recent observations.
Out of the many exponential smoothing techniques available, Holt's method has been chosen for its simplicity and relative ability to deal with heteroscedasticity.

The holt() function from the *fpp2* package fits the model while determining the optimal smoothing parameters alpha (level) and beta (trend). Here are the results of the fitting:

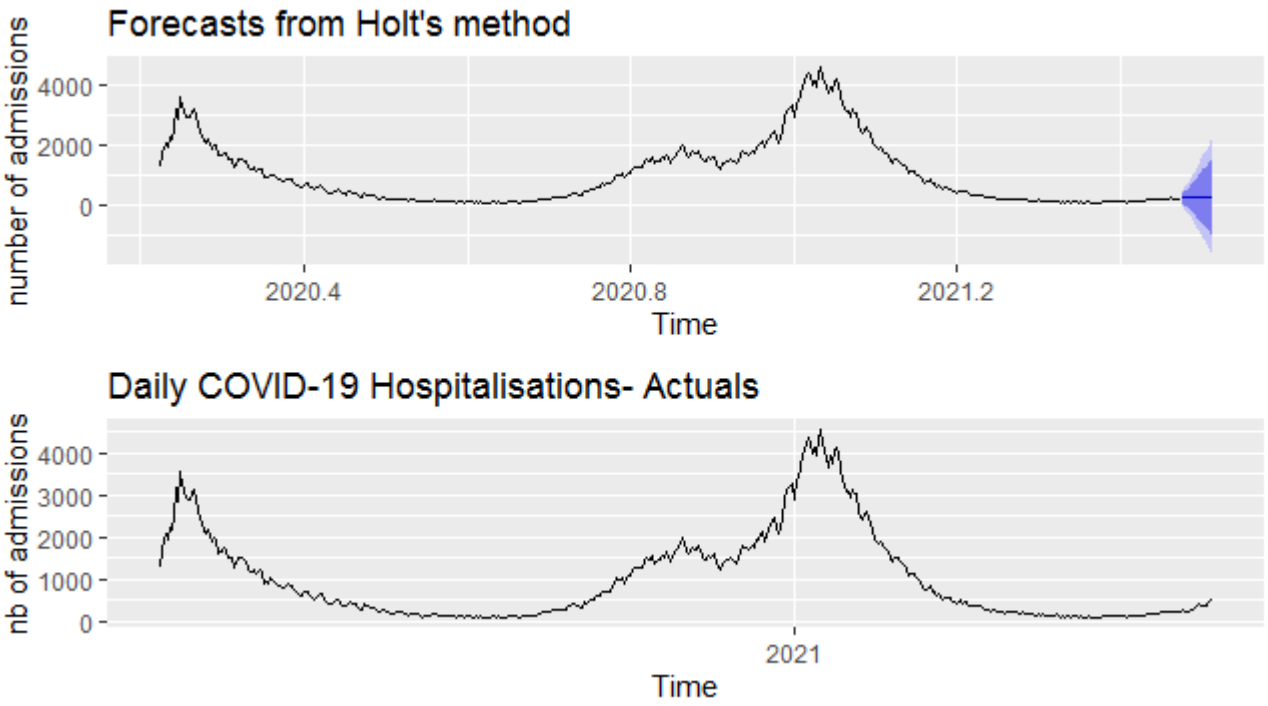
```
> summary(model.holt)

Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = train.ts, h = 14, initial = "simple")

  Smoothing parameters:
    alpha = 0.9266
    beta  = 0.1696

  Initial states:
    l = 1273
    b = 447

  sigma:  124.6691
Error measures:
                    ME     RMSE    MAE         MPE     MAPE        MASE
Training set -6.165822 124.6691 73.004 -0.3730476 9.608049 0.06362848
                    ACF1
Training set -0.002221623
```

Figure 15: Holt's model summary

The holt() function has determined that alpha=0.9266 and beta=0.1696 were optimal values for this time series. The RMSE of the training process is a little higher than for ARIMA(1,1,1) and ARIMA(2,1,2). However, when it comes to forecast, the Holt's method obtained slightly better results, as shown below:

```
> accuracy(model.holt, test.ts)
                     ME     RMSE      MAE         MPE      MAPE        MASE
Training set -6.165822 124.6691 73.0040 -0.3730476  9.608049 0.06362848
Test set     92.000399 123.2120 92.0004 21.4006780 21.400678 0.08018527
                    ACF1 Theil's U
Training set -0.002221623        NA
Test set      0.706664600  2.636194
```

Figure 16: Holt's method forecast accuracy



4. **DISCUSSION**

	RMSE
ARIMA(1,1,1)	144.1743
ARIMA(2,1,2)	148.7777
HOLT'S	123.2120

According to the RMSE, the ES model is the most efficient one for forecasting daily COVID-19 hospitalisations, although the difference between the three models is rather thin.

Many ameliorations could be brought to this analysis for better forecasting, including:

- Finding predictors to perform ARIMAX
- Fine-tuning the parameters alpha & beta of Holt's method
- Comparing the time series with COVID-19 data from other countries.