Introduction:

The below report is regarding the spatial and temporal changes of temperatures for 20 different locations in the United Kingdom for the year 2020. A temperature is a unit used to represent the hotness or coolness of any particle, location, or human body on several scales, including Fahrenheit and Celsius. Data obtained to perform the analysis was from a subset of the UK Met Office's network of land surface measurement sites. We have two csv data files one is metadata.csv which contains the site names, longitude, latitude, and elevation for 20 locations throughout Britain and the other are MaxTemp.csv which contains maximum daily temperatures in degrees Celsius for all the locations for the year 2020, as 2020 is a leap year it has 366 entries per site. Initially, we would be performing spatial analysis for the locations on September 12th 2020, and predict for Morecambe, Coventry, and Kinross location's maximum temperature on this day. Secondly, we would be performing temporal analysis to predict the maximum temperatures in Yeovilton from November 1st to November 7th 2020, and assessing this model's appropriateness for other locations. Respective graphical and numerical summaries of the analysis are presented in the report.

Initial Data Analysis:

Table 3 in the appendix shows that among 20 given locations, the Lowest value of maximum temperatures in a day throughout the year is -2.6 °C and was observed at Dun Fell. The highest value of the maximum temperature recorded was 36.70 °C in London. From the summary table, it can be noted that Dun Fell is the coolest place throughout the year among the given locations because of its elevation.

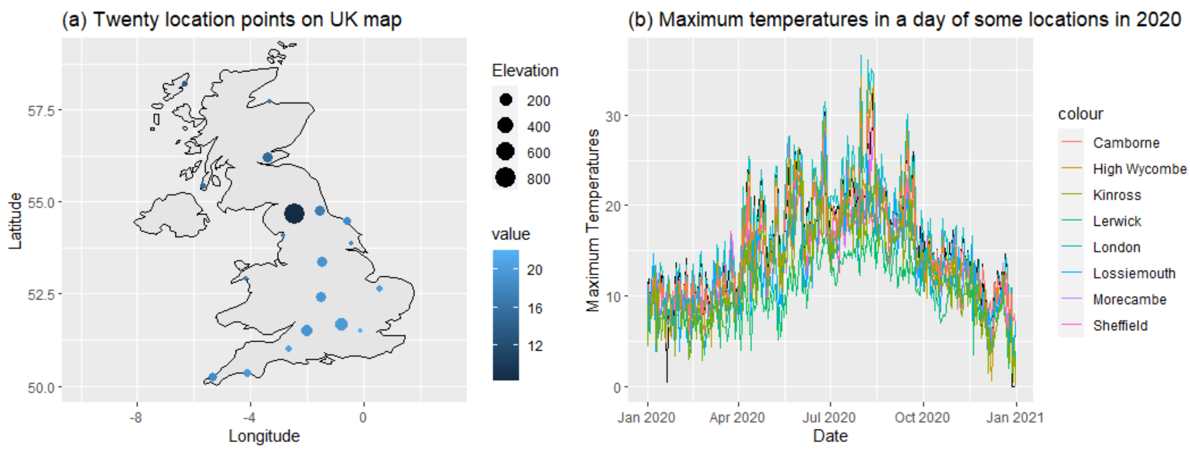


Fig 1: (a) Spatial Locations of all 20 points on the UK map for Sept 12, (b) Temporal series of few locations.

From the above Fig 1b), it can be observed that the maximum temperature values are low between January and March as it is the winter season in the UK, slowly starts increasing from April and having the peak values of maximum temperatures around August till Mid-September. Temperature again starts dipping in October as it is followed by winter. With this we can notice that there is some sort of trend in the data as temperatures start from low values, then increases after a few months, and slowly starts dipping. This makes the relative time series non-stationary. While performing time series analysis we should make it stationary and carry on the next part.

From Fig1a) and Table3, it can be noted that the locations in the north record low maximum temperature compared to the locations in the south in summer and some locations in the mid-UK also record low temperatures in summer comparatively. As we are performing spatial analysis for September 12th, which is summer so the above pattern should be seen on the prediction grid for the model build. And the bigger point is Dun fell, it should record low temperature. Maximum temperatures in a day also depend on elevation, if elevation is large, the temperature is less, and vice-versa.

There are no Missing values in the given data set, so no imputations are required for building the model. Some locations have maximum values as their outlier, and the locations are Dun fell, London, Durham, Porthmadog, Morecambe, Lossiemouth, Yeovilton, Sheffield, Coventry, and Lyneham. (Refer to Fig 11 in Appendix). However these may not affect during modelling due to less difference.

Spatial Analysis on September 12th

A process or group of analytical techniques for determining the spatial correlations between geographical phenomena is known as Spatial modeling. In this part, we would be building a model for locations with the maximum temperatures recorded on this day. And we are supposed to predict the values of Morecambe, Coventry, and Kinross on September 12th 2020. Initially, we would extract the maximum temperature values of all the locations on September 12th. As this day is the 256th day in the year we would be using this number to make the extraction handy. However, we don't have longitude and latitude values of location in this dataset, to get the values of longitude and longitude values we merge the extracted values with the metadata data set, in turn getting coordinates for all locations. As we are supposed to predict values for 3 locations, we remove them from the data and convert them into geodata, making it ready for spatial analysis.

From Fig 12 in the appendix, it can be noted that there is some sort of trend in the data. As you can see in the 1st plot of geodata blue points refer to lower maximum temperature values and red points refer to the highest maximum temperatures on September 12th. So, we need a trend in the model, this can also be supported by the variog4() plot (Fig 13 in the appendix), because it shows that there is directional dependency in the data i.e., anisotropy, so we need to include the trend in the model. For building the spatial model we have the Kriging and Gaussian approach, in the gaussian approach we have likelihood estimation methods, and the bayesian estimation method. Bayesian estimations and the Kriging approach works well for large amounts of data, as we have fewer data points Likelihood approach might be best. Now we build a model with the Matern and Exponential covariance methods with maximum likelihood and restricted maximum likelihood approaches and trend included. Among the build models, the exponential method and matern method with a restricted maximum likelihood (REML) approach have the lowest AIC/BIC and highest loglikelihood values (Table 4 in appendix). However, exponential is not a generally used approach for real data. So, we conclude matern covariance method with a restricted maximum likelihood approach is the best fit method. After finalizing the model, we have to perform validation checks i.e., visualizing residuals and QQ plots.

From the cross-validation plot (Fig 15 in appendix) we can notice that the data is randomly scattered for the predicted and observed, if you look at the histogram of standard residuals and $data - predicted$, it shows that both are normally distributed. And the standard residuals for the data and predicted seem to have constant variance and 95% of the data falls in between $[-1.96, 1.96]$. And the $data{-}predicted$ values for data and predicted also seem randomly scattered. A strong relationship holds between true and predicted data.

Now let's visualize the prediction grid with the predicted location points on the grid. Here we will be visualizing the Mean and variance on the 0.1^{o} prediction grid. As we already have actual values of three locations it would be better to compare them with the values on the prediction grid. Below table1 contains the actual values of temperatures for predicted locations on 12th September and compares them with values on the prediction grid.

From the comparison of Fig 2a and table1, it can be noticed that the predicted values have maximum temperature values almost equal to the actual. And if you look at the uncertainties grid it can be observed that at the prediction points variance is high and at the observed values variance is low. This can be observed in the plot of observed values in the appendix. As the prediction for 3 locations is good, the above built model is good for the spatial approach.

Table 1: Longitude, Latitude, and Actual maximum temperature values on September 12th, 2020

Location	Longitude	Latitude	Maximum Temperature
Coventry	-1.536	52.424	19.6
Kinross	-3.413	56.214	15.0
Morecambe	-2.860	54.076	17.3

a) b)

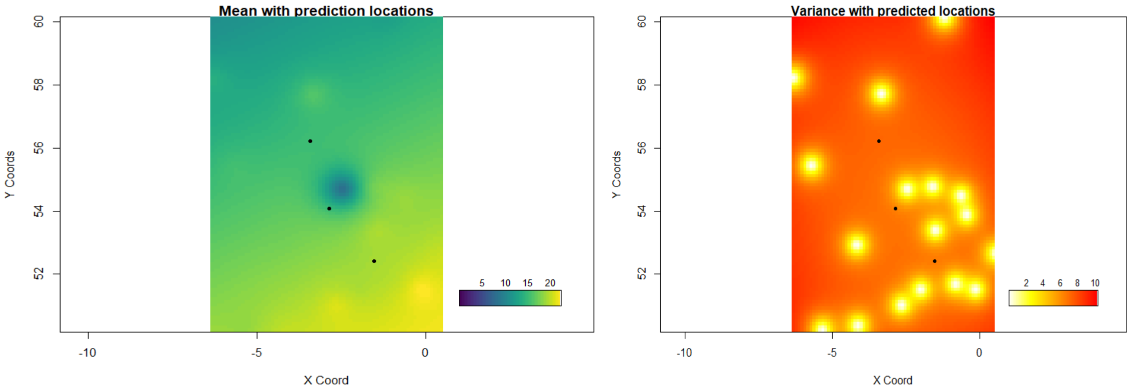


Fig 2: a) Map of predicted temperatures b) Map of uncertainties on a 0.1degree grid.

Spatial Analysis of the UK in Spring, Summer, and Winter:

In the UK, Spring is from March to May, Summer is from June to August, Autumn is from September to November and Winter is from December to February. So now we would perform spatial analysis of 1 day in Spring i.e., 26th April 2020, 1 day in winter i.e., 31st December 2020, and 1 day in summer i.e., 05th July 2020. Initially, we would be taking maximum temperatures of all these three dates separately, and converting them into geodata, performing spatial analysis. As it is mentioned earlier kriging approach and Bayesians approach in the gaussian process works well for big data. And for the above analysis for September 12, Matern covariance method with the Restricted likelihood method was a good approach, so we would be carrying analysis on the same method.

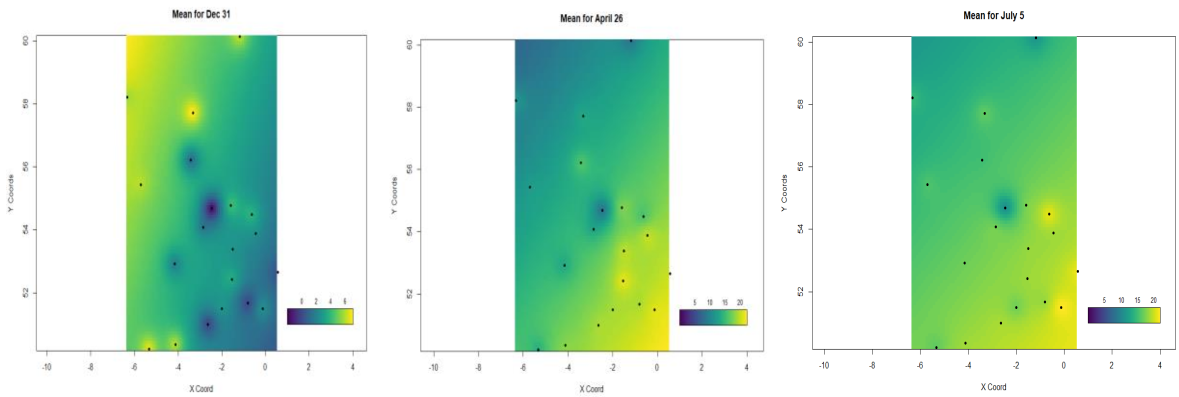


Fig 3: Spatial Analysis of 20 UK Locations in Winter, Spring, and Summer on 0.1° grid

From the cross-validation plots (Fig 16-18 in appendix) for the above three models, we can notice that the data is randomly scattered for the predicted and observed, if you look at the histogram of standard residuals and *data − predicted*, it shows that both are normally distributed. And the standard residuals for the data and predicted seem to have constant variance and 95% of the data falls in between [−1.96, 1.96]. And the *data−predicted* values for data and predicted also seem randomly scattered. A strong relationship holds between true and predicted data.

From the above three prediction grids, it can be noted that overall temperatures in winter i.e., December 31st are low as in winter, and in spring most of the locations in the north have low temperatures relatively, and in summer it seems like temperatures of locations are increased. And in all three grids, the middle blue point refers to the Dun Fell location, as its elevation is higher it stays cool throughout the year when compared to other locations.

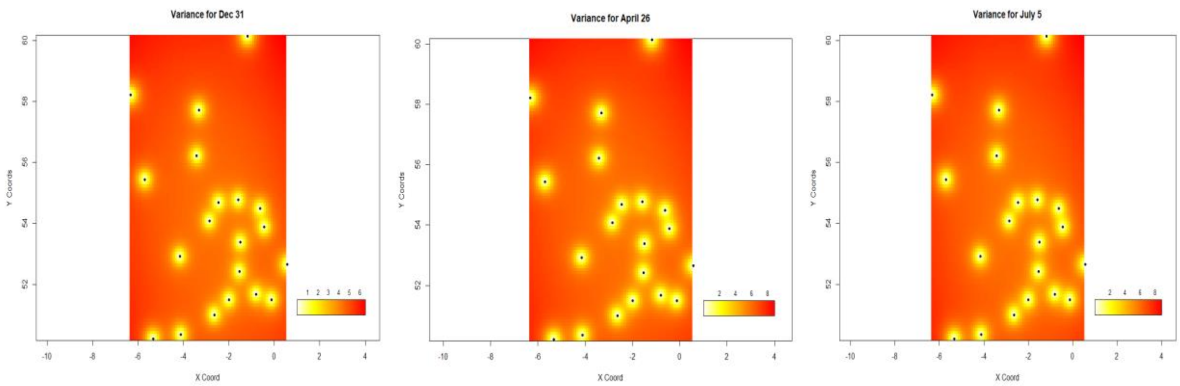


Fig 4: Map of uncertainties in Winter, Spring, and Summer

Fig 4 shows the variance of December 31st, April 26th, and July 5th data, it follows the pattern as the variance moves away from the data it increases.

Time series Analysis for Yeovilton from 1st November 2020 to 7th November 2020

For performing the time series analysis, we initially convert the data into time series data using the ts() function. Here we are supposed to perform an analysis and predict the maximum temperature in a day for Yeovilton in between 1st and 7th November 2020. For this, we will be building the model with values from 1st January 2020 to 31st October 2020 on daily basis, and later on, predict for the next seven days. Initially, we visualize the time series of Yeovilton. From Fig 5a) it can be observed that the time series is non-stationary i.e., there is some sort of trend in the model, and the trend for this data might definitely be the quadratic trend as it first increases and then slowly starts decreasing. And from Fig 5a) and from general theory, it can be concluded that there is no seasonality for data on a daily basis, so we would be building ARIMA model for the data. For this, we have to check how much differencing is required to make the data stationary time series. Fig 5b) shows 1st differencing of the time series stationary, this can also be supported by the *ndiff()* function which returns 1 differencing is needed for the data.

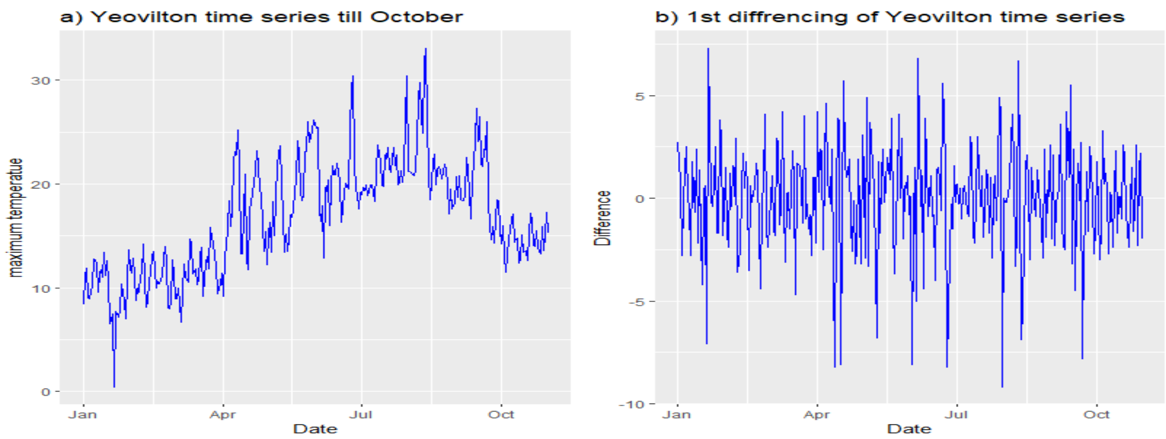


Fig 5: a) Yeovilton time series till October, b) Difference of Yeovilton time series.

Looking at ACF and PACF(Fig20)it is unable to clearly decide the order. After this, we run auto. arima() with maximum p, q, and d values as 4 each. This function gives the best order for the values most of the time, however, sometimes we need to play with the order around the given values if coefficients are not significant or residuals are auto-correlated, etc., Here auto.arima() returned ARIMA(2,1,1) is the best model for the data, however, coefficients contain zero in their confidence interval making them not significant. After changing the order and performing appropriate checks ARIMA(1,1,1) and ARIMA(1,1,2) have good residuals and significant coefficients, but ARIMA(1,1,2) has less AIC value, good σ^2 = 5.933 and log-likelihood = -702.35, so we conclude this as our final model.

Significance of coefficients:

For ar1: [0.5393-1.96*0.0935, 0.5393+1.96*0.0935] = [0.3560, 0.7225]

For ma1: [-0.7237-1.96*0.1020, -0.7237+1.96*0.1020] = [-0.9236, -0.5237]

For ma2: [-0.1716-1.96*0.0796, -0.1716+1.96*0.0796] = [-0.3276, -0.0155]

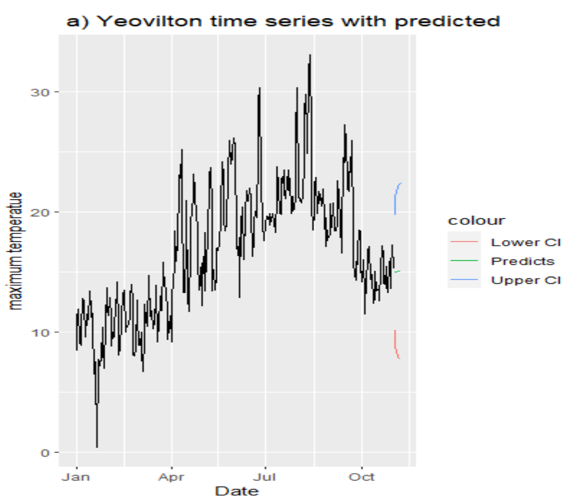
All the above coefficients do not contain zero in their confidence interval, in turn making them significant.

Residual Checking:

After visualizing residuals, it can be noted that there is no auto-correlation in the residuals, and almost 95% of the residuals fall in between normal intervals i.e., $[-1.96, 1.96]$. Here the residuals look like white noise. The acf plot shows no autocorrelation between the residuals, which is confirmed by the Ljung-Box statistic (Fig 21 in appendix). And even the coefficients are significant. After the respective validations of models, we are predicting the maximum temperature values of Yeovilton from 1st November 2020 to 7th November 2020.

Table 2: Actual, Predicted, 80% Interval and 95% interval of Yeovilton from 1st to 7th November

Date	Actual	Predicted	80% Lower	80% Upper	95% lower	95% Upper
01-11-2020	17.3	14.93705	11.81538	18.05872	10.162863	19.71124
02-11-2020	16.5	14.99371	10.96534	19.02209	8.832848	21.15458
03-11-2020	11.1	15.02427	10.65187	19.39667	8.337261	21.71128
04-11-2020	10.7	15.04075	10.49491	19.58659	8.088485	21.99302
05-11-2020	12.0	15.04964	10.39562	19.70366	7.931932	22.16735
06-11-2020	12.9	15.05443	10.32074	19.78813	7.814867	22.29400
07-11-2020	14.8	15.05702	10.25731	19.85673	7.716497	22.39754



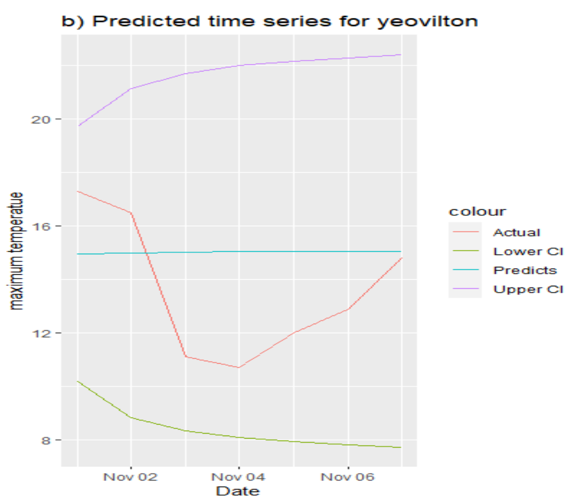


Fig 6: a) Yeovilton time series with actual and predicted values, b) Predicted values series of Yeovilton.

Root Mean Squared Error is used to measure the differences in actual values and predicted values, for this model, RMSE is 2.831. And from the above table, 95% confidence interval contains data so it's good model. For the appropriateness of this, the same model was built for London and Sheffield locations, these models also have good residuals and no auto-correlation (Fig 22 in appendix) , and even the coefficients are significant for these models (refer to model3 and model 4). And we made predictions from 1st November 2020 to 20th November 2020, predictions were also good and even the 95% CI captures the data. (Table5,6)

Extension of Time series Analysis:

Here we would be performing two types of analysis 1) we would be building model for the daily data and predicting for the next 14 days of any desired location 2) we would be averaging data on a weekly basis and predicting for the next weeks of the same locations and compare them.

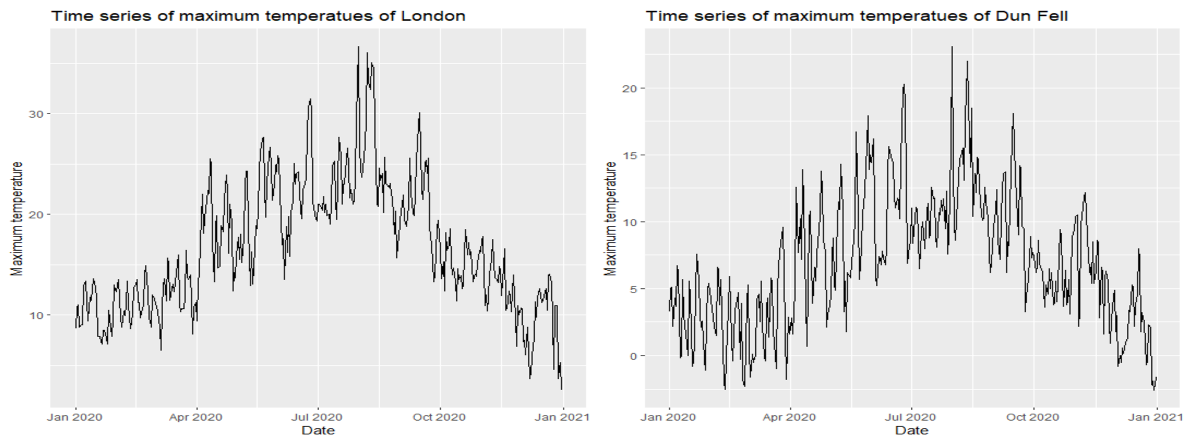


Fig 7: a) Time series of Maximum temperatures of London b) Dun fell in 2020.

As both time series look non-stationary, we would be building ARIMA models. As part of the building model, we already performed appropriateness of ARIMA(1,1,2) for other locations, here initially I had built the models in the same order. This model has given good residuals with white noise and no auto-correlation (Fig 23). And even coefficients were significant for both models (model 5 &6). However, because of changes in data AIC, log-likelihood and σ^2 values differ. So, for both location ARIMA(1,1,2) model holds good. Now we would be predicting for the next 14 days i.e., from 1st January 2021 to 14th January 2021. As January is winter in the UK, the temperatures of any location would be very low and the predictions also show the temperatures are low for the locations(Table 7& 8 in appendix), so we can conclude that predictions are good.

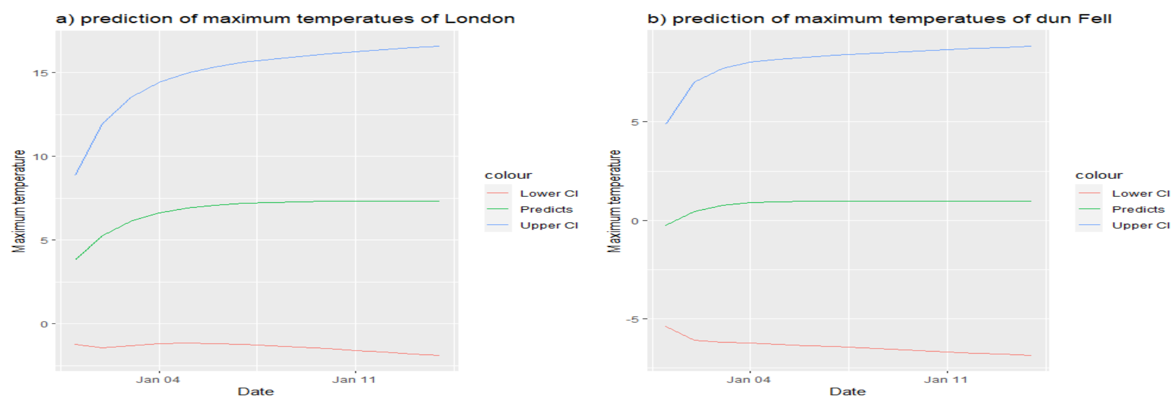


Fig 8: a) Prediction of daily maximum temperatures of London from 1st January to 14th January b) prediction of daily maximum temperatures of Dun Fell.

Now we would be performing analysis on weekly data, for this, we need to introduce the week in the data frame with the help of the date and later on average the maximum temperatures on weekly basis for all the locations. Now we will be having 53 weeks with average maximum temperatures for all locations. And we would be performing analysis on weekly basis here. Time series is non-stationary for a weekly basis as well, refer to fig 9.

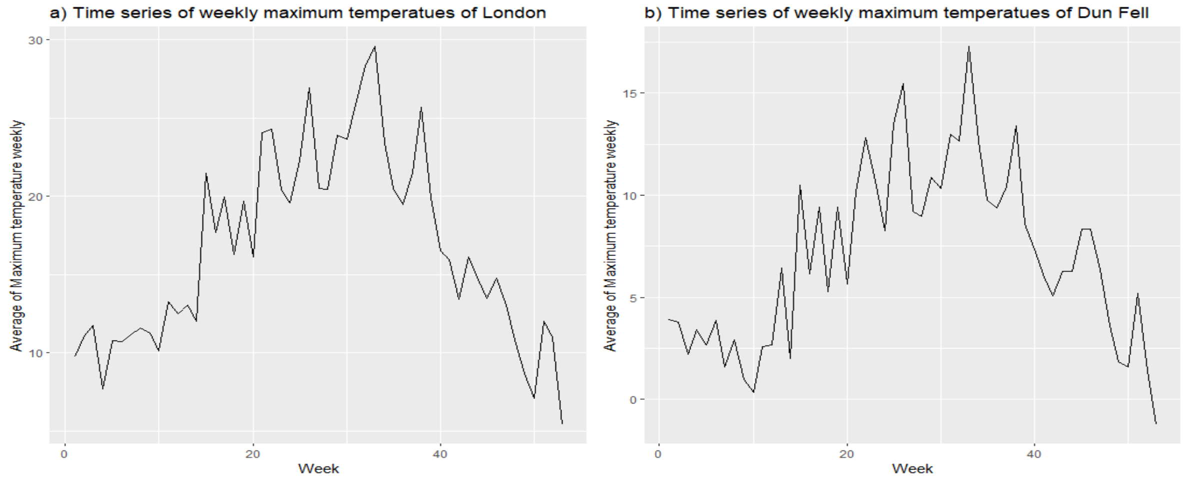


Fig 9: a) Time series of weekly average time series of London in 2020 b) Time series of weekly average time series of Dun fell in 2020.

For this, we need to build the model with the function auto.arima() for London weekly data, this has given ARIMA(2,1,2) (model7) which has significant coefficients and good residuals with white noise and no auto-correlation(Fig 26). Later, we need to perform appropriateness of the model, after performing for some locations it has given significant coefficients and good residuals, so ARIMA(2,1,2) is the best model for weekly average maximum temperature data. After this, we would be predicting the next 4 weeks for both London and Dun Fell with respective models (model7&8) and Fig 10 shows the predictions along with Confidence Intervals. As these 4 weeks are from the start of January and might be till January end, slowly winter season would be coming to end so temperatures have to increase weekly and this can be seen in (table 9&10 in the appendix) and Fig 10.

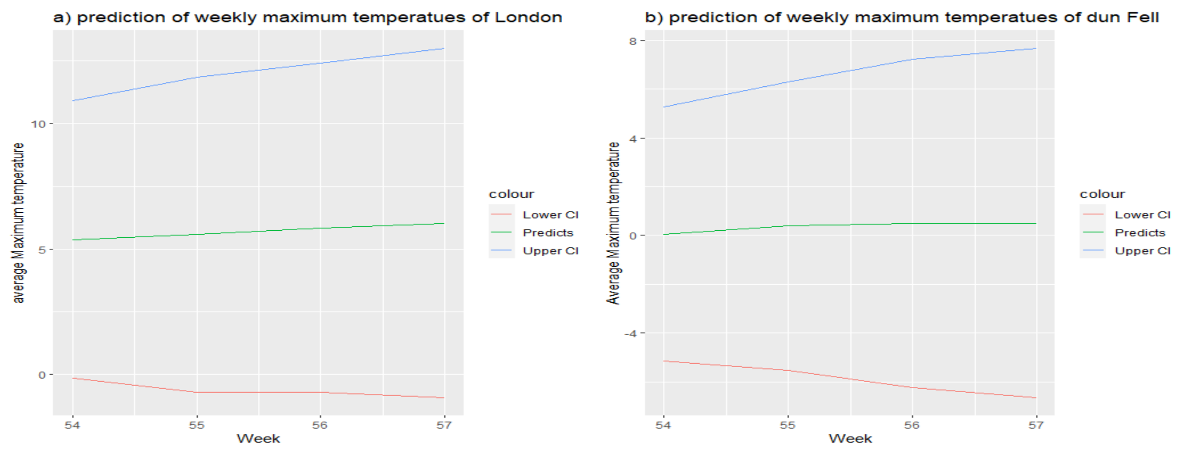


Fig 10: a) Predictions of weekly average time series of London b) Predictions of weekly average time series of Dun fell.

If you look at the predictions on a daily data basis and weekly data basis, Daily data basis temperature predictions have become high after 4 or 5 days and stay at high temperatures comparatively and stay at that particular temperature till the end of predictions. For weekly predictions, it can be concluded that the overall average maximum temperature in London in January is around 5.5 ^{o}C and for Dun fell around 0.5^{o}C. When the data is averaged every week there might be some loss in data. So, predictions might differ. And there might be a change of dates when it melted into weeks for eg; the last week of 2020 might have two or three days, because of this lag in data points it's not good to compare with daily data predictions averaged for 7 days with the week. Refer to table 7,8,9,10 in the appendix for daily and weekly predictions for London and Dun fell locations.

Conclusion:

In the above report, we initially spoke about Initial data analysis and places that recorded the highest and lowest temperatures of the UK in 2020 there we spoke about missing values and outliers in the data. Secondly, we performed spatial analysis on 12th September, here we built the spatial model for 17 locations and predicted the maximum temperatures on 12th September 2020 of Morecambe, Kinross, and Coventry, then compared with the observed values and predictions seems almost good because it captures nearly observed maximum temperatures. Then, we extended the spatial analysis for one day in winter i.e., December 31st, one day in summer i.e., July 5th, and one day in spring (April 26th), in these models the built prediction grid shows mean images based on seasons i.e., cool temperatures in winter, high temperatures in summer and moderate in spring. Finally, we build a temporal model for Yeovilton and predicted maximum temperature in a day from 01st November 2020 to 07th November 2020, as the Yeovilton time series was not stationary, we built the Arima model for the data. Then we checked the appropriateness of the model with the London and Sheffield locations. Later on, we extended temporal analysis for this we initially build a model for London and Dun fell locations for 2020 data and predicted temperatures from 01st January 2021 to 14th January 2021, then we averaged the data of 2020 by week and built the temporal model on weekly basis, then predicted the weekly average data for 4 weeks in 2021 for London and Dun fell locations, due to some loss of data during averaging predictions might differ from the average of daily basis predictions for 1st two weeks.

Appendix

Table 3: Summary statistics table of MaxTemp data set.

Location	**Minimum**	**1st Quartile**	**Median**	**Mean**	**3rd Quartile**	**Maximum**
Machrihanish	4.7	9.7	12.6	12.86	15.50	23.30
High Wycombe	0.0	9.925	13.600	14.87	19.875	34.20
Camborne	6.6	11.1	13.5	14.17	17.2	25.5
Dun Fell	-2.6	3.425	6.60	7.042	10.4	23.1
Plymouth	5.0	11.4	14.6	15.3	19.2	29.1
Durham	0.0	9.9	13.5	13.82	17.5	30.8
London	2.5	11.7	15.75	16.75	21.38	36.70
Porthmadog	2.0	10.1	14.2	14.56	17.9	30.2
Leconfield	2.5	9.85	13.7	14.39	18.40	30.30
Kinross	0.4	8.60	12.7	12.69	16.57	26.4
Morecambe	1.6	10.0	13.4	13.95	17.38	31.1
Lossiemouth	1.5	9.025	12.2	12.928	16.375	28.6
Marham	0.1	10.43	14.30	15.31	20.18	34.10
Whitby	3.80	9.8	13.0	13.94	17.82	29.6
Lerwick	1.9	7.6	9.9	10.17	13.00	18.40
Yeovilton	0.00	11.50	15.15	15.83	20.10	33.10
Sheffield	2.7	9.925	13.7	14.51	18.9	34.3
Coventry	1.9	10.43	14.10	15.10	19.77	35.10
Stornoway	3.4	8.80	11.15	11.70	14.68	23.00
Lyneham	2.20	10.6	13.95	14.88	19.57	33.2

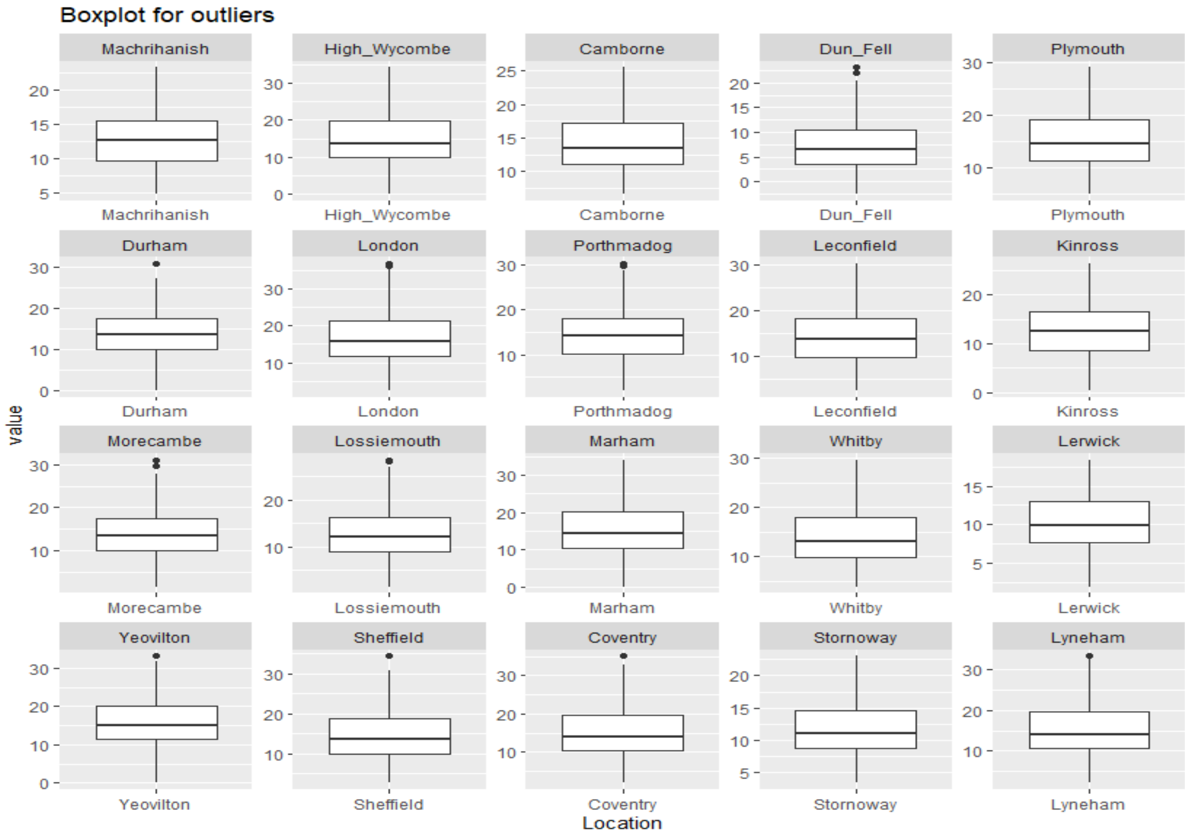


Fig 11: Boxplot for outliers

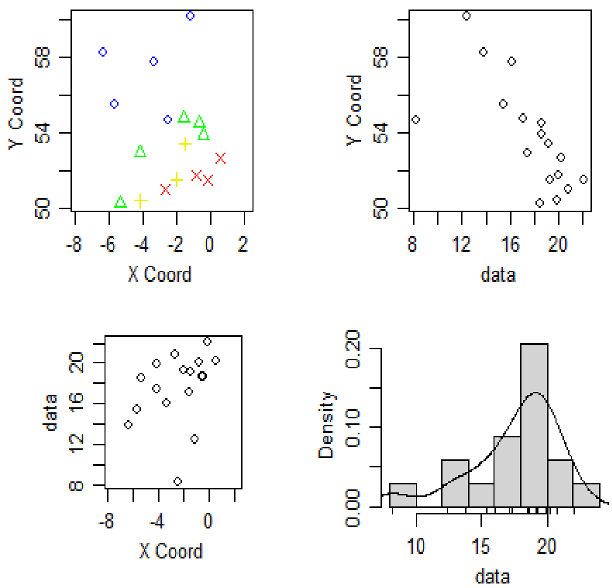


Fig 12: Geodata plot for September 12 2020

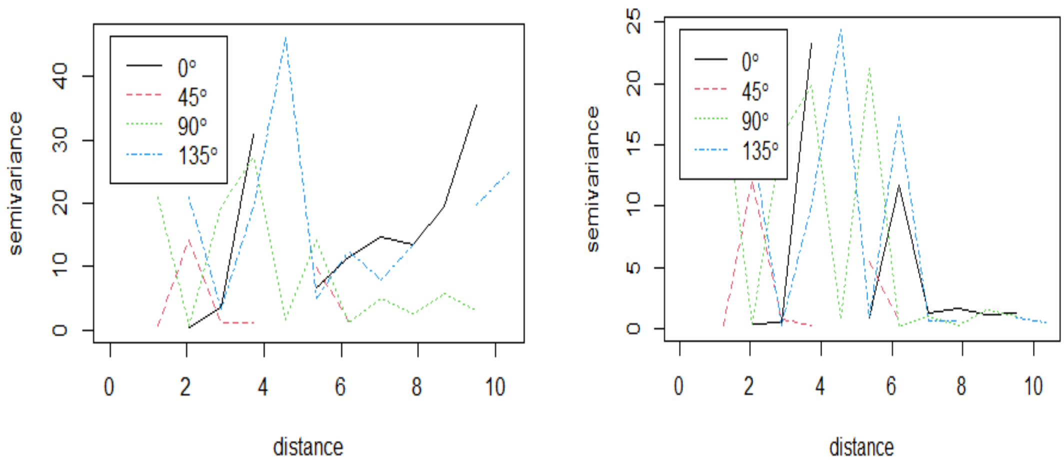


Fig 13: Variog4 plot with the constant trend and linear trend

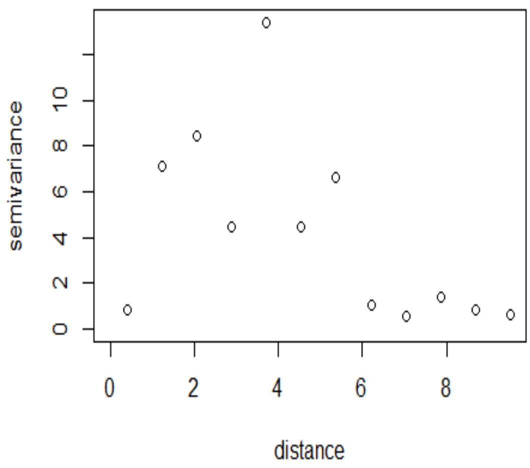


Fig 14: Variogram plot for September 12th, 1st trend

Table 4: Covariance methods maximized likelihood estimates for September 12th 2020 model

Method	loglikelihood	AIC	BIC
Matern ML	-38.12	88.24	93.24
Matern REML	-32.65	77.3	82.3
Exponential ML	-38.12	88.25	93.25
Exponential REML	-32.66	77.32	82.32

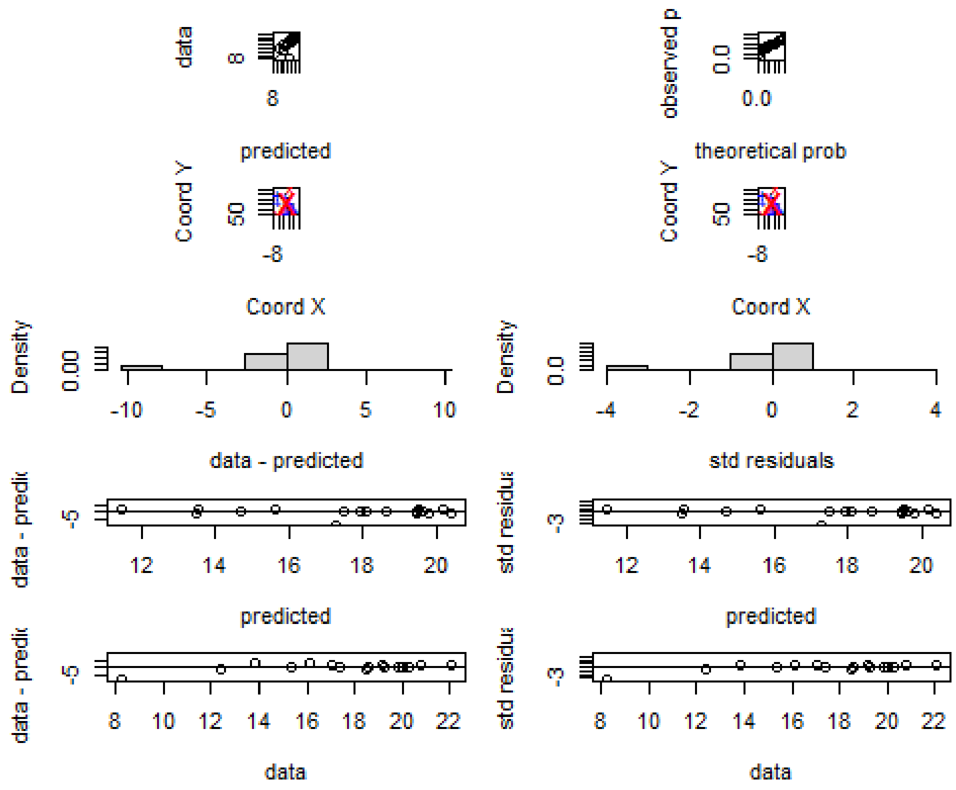


Fig 15: Cross-validation plot for September 12th data for REML matern method.

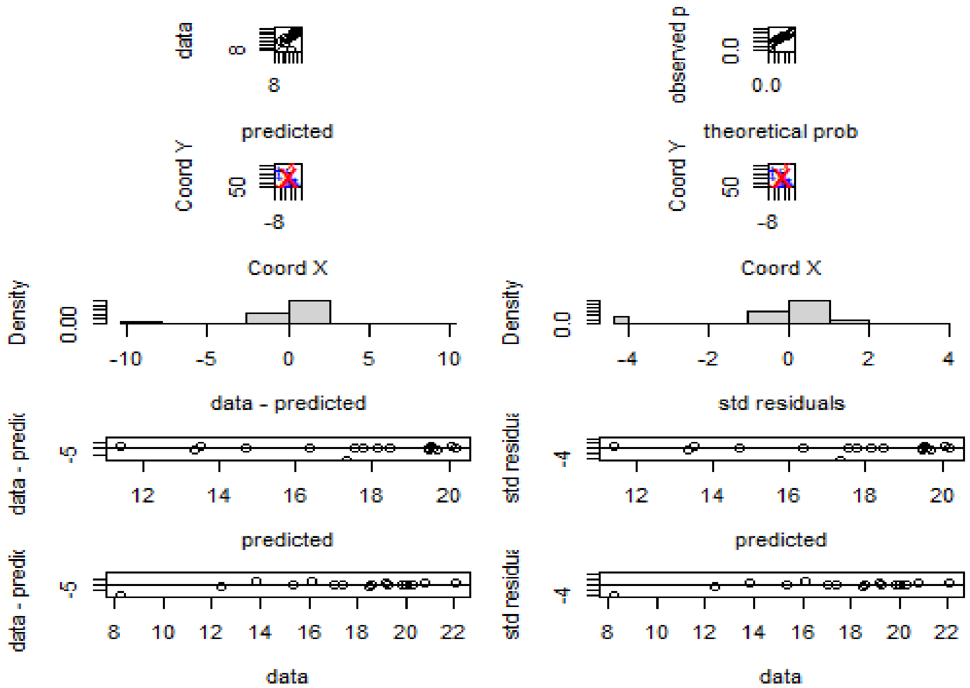


Fig 16: Cross-validation plot for December 31st data for REML matern method.

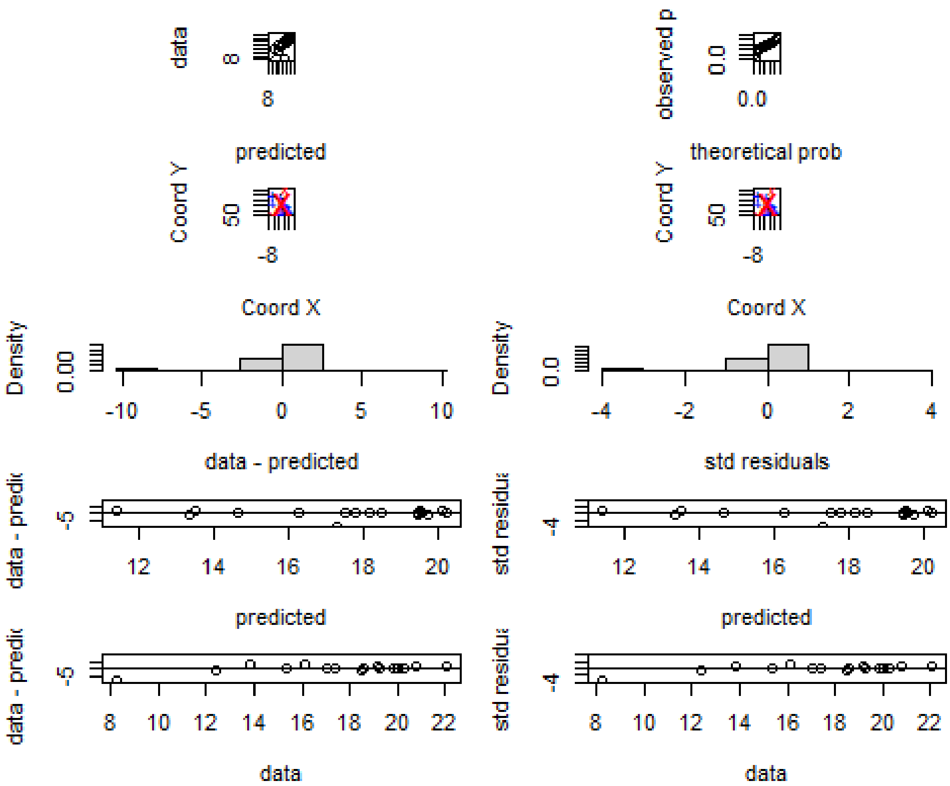


Fig 17: Cross-validation plot for April 26th Data, REML matern method.

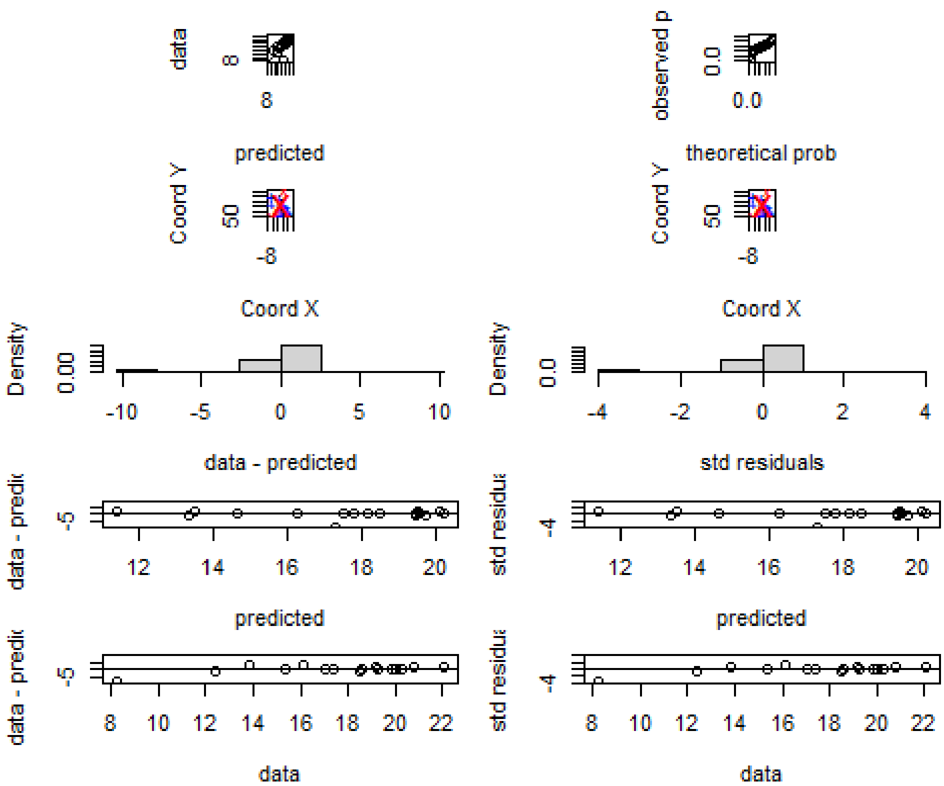


Fig 18: Cross-validation plot for July 5th 2020, REML matern

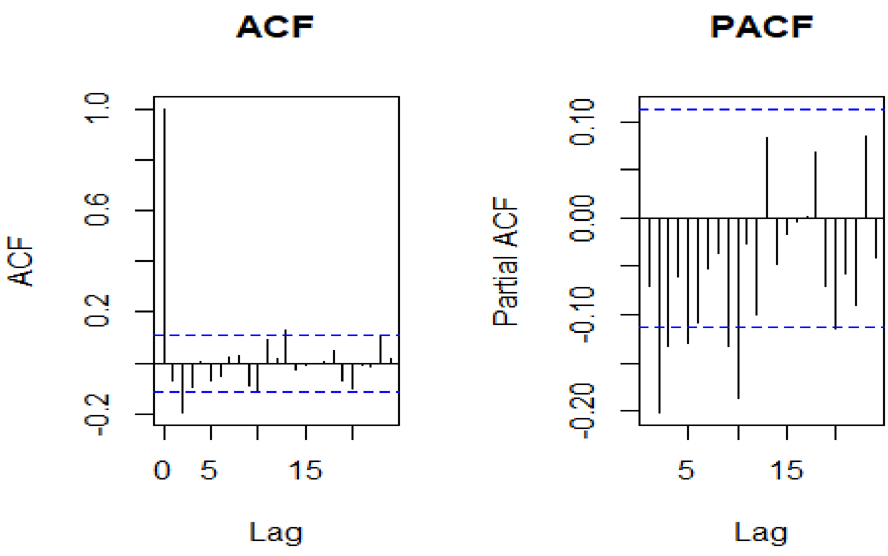


Fig 20: ACF and PACF plot of differenced Yeovilton data.

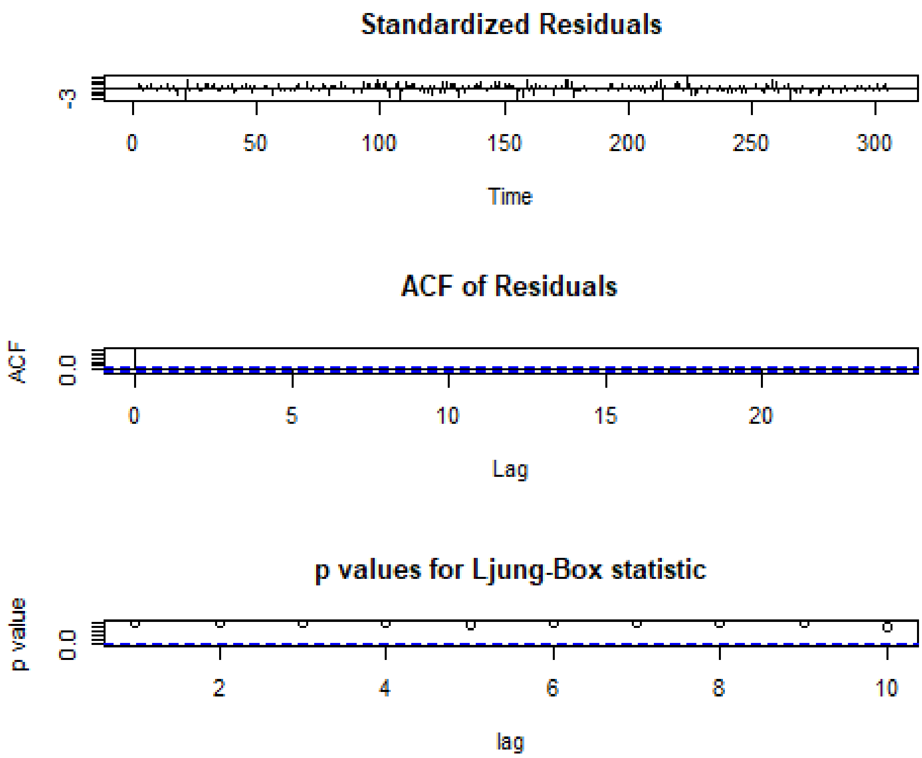


Fig 21: Residuals plot for Yeovilton model ARIMA(1,1,2)

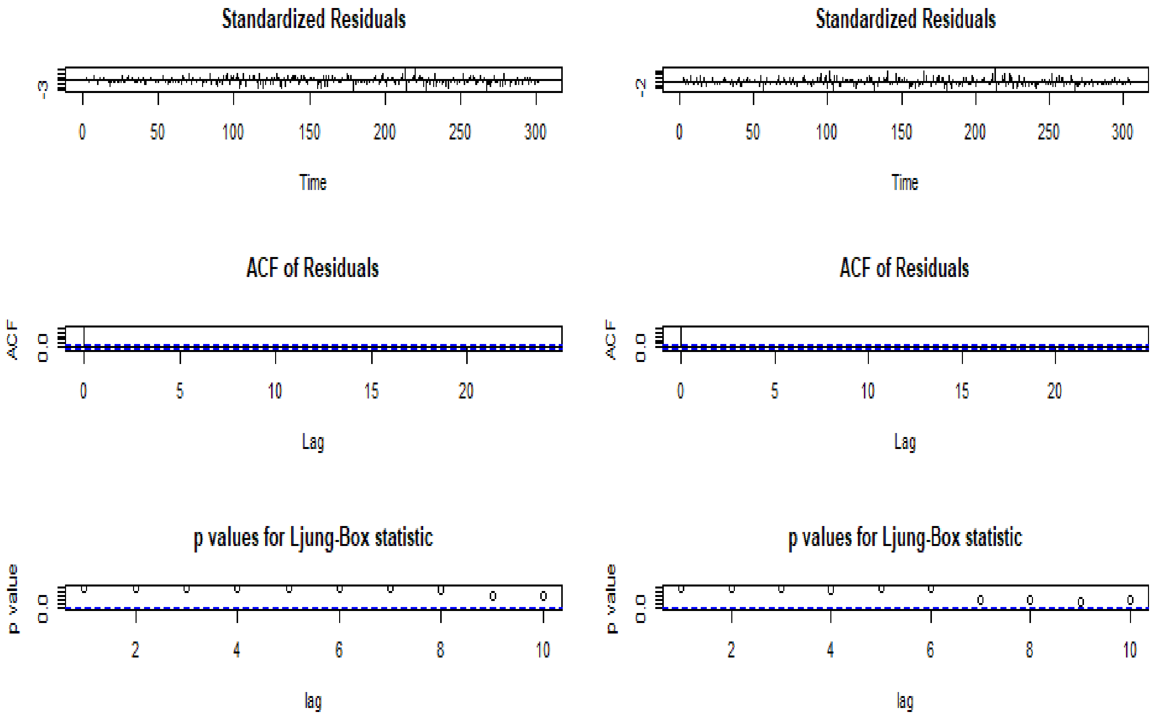


Fig 22: a) Residuals for London model for appropriateness; b) residuals for Sheffield model for appropriateness

Table 5: Table of Prediction, 80 and 95% CI and actual values for london 1st to 20th November

Date	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95	actual
2020-11-01	16.03445	12.732085	19.33681	10.983921	21.08497	17.4
2020-11-02	15.86344	11.453388	20.27350	9.118846	22.60804	17.8
2020-11-03	15.75719	10.853943	20.66044	8.258321	23.25606	10.9
2020-11-04	15.69117	10.514950	20.86738	7.774827	23.60751	12.0
2020-11-05	15.65014	10.298825	21.00146	7.466008	23.83428	10.4
2020-11-06	15.62465	10.146780	21.10253	7.246969	24.00234	11.7
2020-11-07	15.60881	10.030377	21.18725	7.077330	24.14030	14.0
2020-11-08	15.59897	9.934720	21.26323	6.936245	24.26170	15.5
2020-11-09	15.59286	9.851567	21.33415	6.812311	24.37341	17.5
2020-11-10	15.58906	9.776177	21.40194	6.699023	24.47909	15.2
2020-11-11	15.58670	9.705749	21.46764	6.592564	24.58083	13.7
2020-11-12	15.58523	9.638600	21.53186	6.490644	24.67982	13.5
2020-11-13	15.58432	9.573701	21.59494	6.391872	24.77677	13.2
2020-11-14	15.58375	9.510418	21.65709	6.295389	24.87212	14.8
2020-11-15	15.58340	9.448356	21.71844	6.200660	24.96614	14.5

Date	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95	actual
2020-11-16	15.58318	9.387267	21.77910	6.107349	25.05901	11.9
2020-11-17	15.58305	9.326991	21.83910	6.015236	25.15085	14.8
2020-11-18	15.58296	9.267424	21.89850	5.924181	25.24174	16.6
2020-11-19	15.58291	9.208496	21.95732	5.834086	25.33173	10.5
2020-11-20	15.58288	9.150159	22.01559	5.744885	25.42087	10.6

Table 6: Table of Prediction, 80 and 95% CI and actual values for Sheffield 1st to 20th November

Date	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95	actual
2020-11-01	13.30975	9.635038	16.98446	7.689764	18.92973	17.3
2020-11-02	13.24621	8.585623	17.90680	6.118457	20.37396	17.3
2020-11-03	13.21730	8.252194	18.18241	5.623826	20.81077	8.5
2020-11-04	13.20415	8.099178	18.30912	5.396771	21.01152	10.8
2020-11-05	13.19816	8.005960	18.39036	5.257374	21.13895	11.1
2020-11-06	13.19544	7.935884	18.45500	5.151643	21.23924	9.5
2020-11-07	13.19420	7.875713	18.51269	5.060275	21.32813	8.0
2020-11-08	13.19364	7.820123	18.56715	4.975556	21.41172	9.1
2020-11-09	13.19338	7.766835	18.61993	4.894196	21.49257	12.8
2020-11-10	13.19326	7.714843	18.67168	4.814742	21.57179	13.3
2020-11-11	13.19321	7.663688	18.72273	4.736536	21.64989	12.6
2020-11-12	13.19319	7.613161	18.77321	4.659273	21.72710	11.4
2020-11-13	13.19318	7.563157	18.82319	4.582805	21.80355	10.9
2020-11-14	13.19317	7.513626	18.87271	4.507057	21.87928	13.9
2020-11-15	13.19317	7.464538	18.92180	4.431984	21.95435	11.5
2020-11-16	13.19317	7.415873	18.97046	4.357559	22.02878	10.6
2020-11-17	13.19317	7.367618	19.01872	4.283759	22.10257	14.9
2020-11-18	13.19317	7.319761	19.06657	4.210568	22.17577	14.1
2020-11-19	13.19317	7.272291	19.11404	4.137969	22.24836	8.6
2020-11-20	13.19317	7.225199	19.16113	4.065948	22.32039	11.2

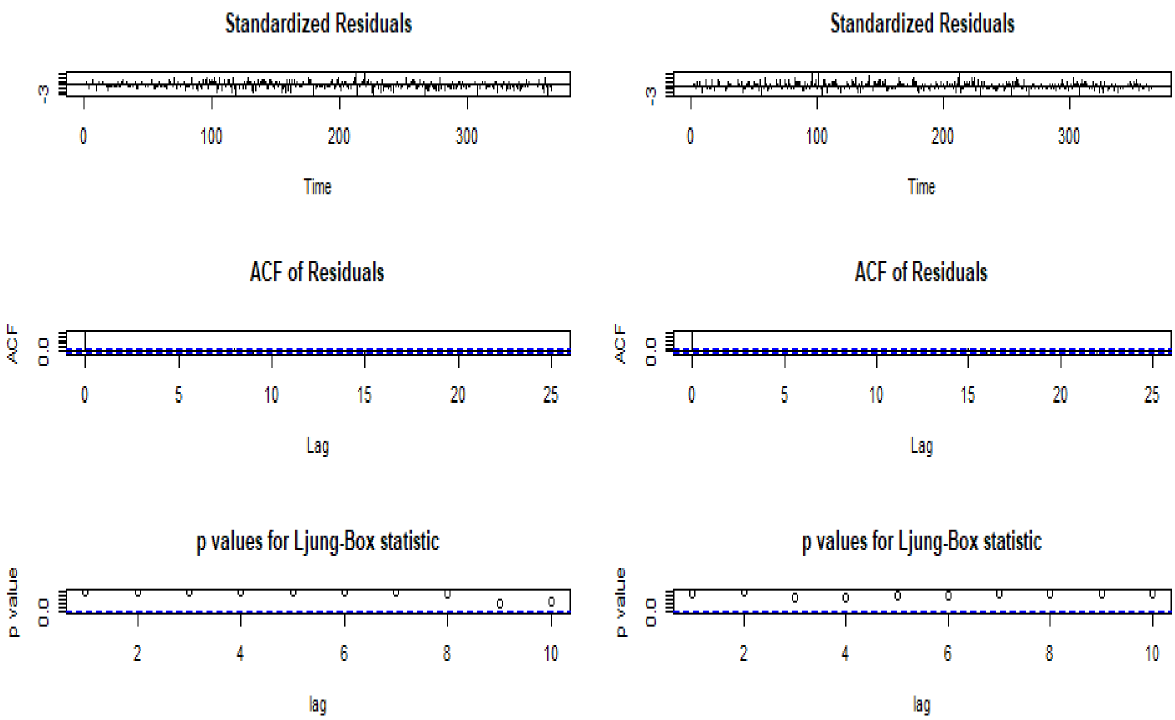


Fig 23: Residuals for London and Dun fell models for 2021 predictions

Table 7: Table of Prediction, 80 and 95% CI values for london from 1st to 14th January 2021

Date	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2021-01-01	3.816391	0.5349385	7.097844	-1.202157	8.83494
2021-01-02	5.250696	0.8926506	9.608741	-1.414359	11.91575
2021-01-03	6.101095	1.2757658	10.926425	-1.278609	13.48080
2021-01-04	6.605297	1.5218092	11.688785	-1.169227	14.37982
2021-01-05	6.904238	1.6523045	12.156173	-1.127901	14.93638
2021-01-06	7.081481	1.7043690	12.458593	-1.142102	15.30506
2021-01-07	7.186568	1.7069220	12.666214	-1.193827	15.56696
2021-01-08	7.248874	1.6793144	12.818434	-1.269032	15.76678
2021-01-09	7.285815	1.6337619	12.937869	-1.358254	15.92989
2021-01-10	7.307718	1.5777637	13.037672	-1.455491	16.07093
2021-01-11	7.320704	1.5158534	13.125554	-1.557049	16.19846
2021-01-12	7.328403	1.4507463	13.206060	-1.660697	16.31750
2021-01-13	7.332968	1.3840567	13.281880	-1.765107	16.43104
2021-01-14	7.335675	1.3167382	13.354611	-1.869494	16.54084

Table 8: Table of Prediction, 80 and 95% CI values for Dun Fell from 1st to 14th January 2021

Date		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2021-01-01	367	-0.2508996	-3.597995	3.096196	-5.369840	4.868041
2021-01-02	368	0.4516715	-3.826812	4.730155	-6.091704	6.995047
2021-01-03	369	0.7563598	-3.786033	5.298752	-6.190630	7.703349
2021-01-04	370	0.8884958	-3.769580	5.546572	-6.235417	8.012409
2021-01-05	371	0.9458001	-3.783451	5.675052	-6.286966	8.178566
2021-01-06	372	0.9706516	-3.813770	5.755073	-6.346490	8.287793
2021-01-07	373	0.9814290	-3.851664	5.814522	-6.410149	8.373007
2021-01-08	374	0.9861030	-3.892758	5.864964	-6.475471	8.447677
2021-01-09	375	0.9881300	-3.935053	5.911313	-6.541228	8.517488
2021-01-10	376	0.9890090	-3.977665	5.955683	-6.606863	8.584881
2021-01-11	377	0.9893902	-4.020211	5.998991	-6.672133	8.650914
2021-01-12	378	0.9895556	-4.062528	6.041639	-6.736940	8.716051
2021-01-13	379	0.9896273	-4.104550	6.083805	-6.801245	8.780499
2021-01-14	380	0.9896584	-4.146253	6.125570	-6.865040	8.844357

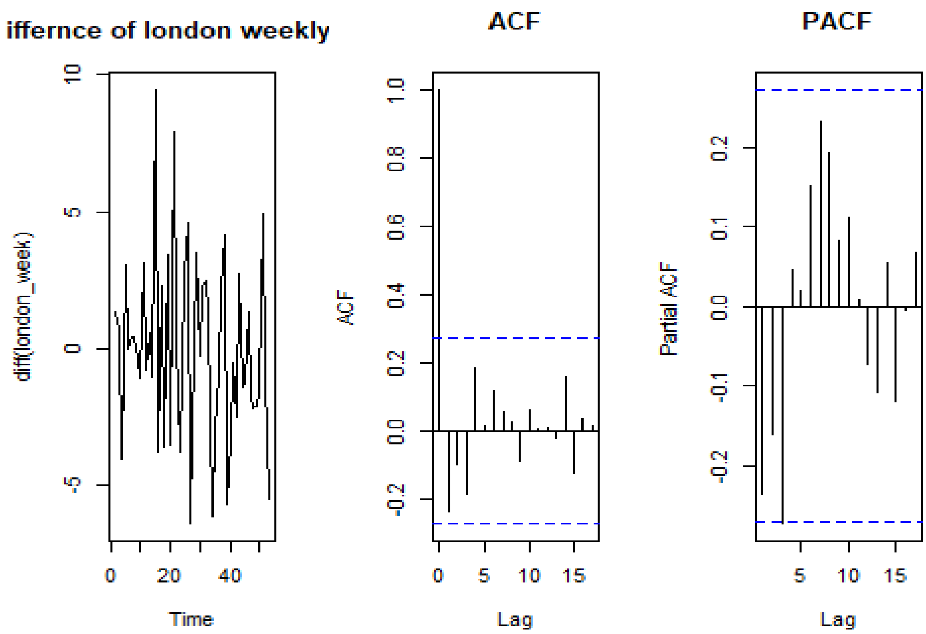


Fig 24: Plot of differenced time series of london, ACF and PACF of corresponding data.

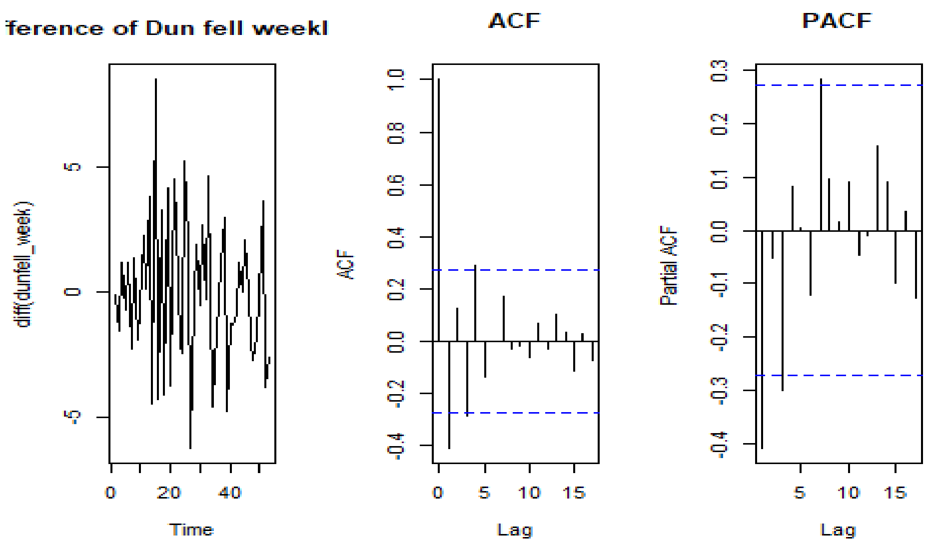


Fig 25: Plot of differenced time series of Dun fell, ACF, and PACF of corresponding data.

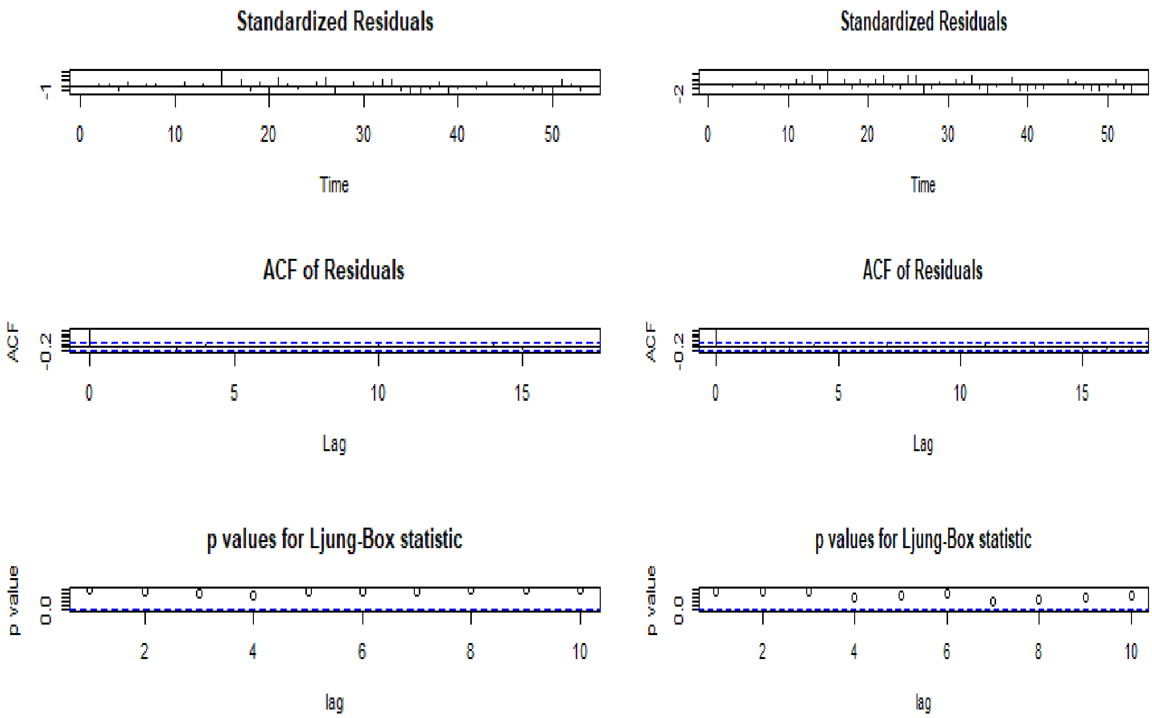


Fig 26: Residuals plot for a)London and b)Dun fell weekly data models.

Table 9: Weekly predictions and 80 and 95% CI for London

Week	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
54	5.367982	1.595257	9.140708	-0.4019032	11.13787
55	5.575564	1.299063	9.852065	-0.9647793	12.11591
56	5.845093	1.364506	10.325681	-1.0073743	12.69756
57	6.033746	1.280237	10.787255	-1.2361188	13.30361

Table 10: Weekly predictions and 80 and 95% CI for Dun fell

Week	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
54	0.0619590	-3.343650	3.467569	-5.146471	5.270389
55	0.3905454	-3.471724	4.252815	-5.516286	6.297377
56	0.4961116	-3.901278	4.893501	-6.229115	7.221338
57	0.5109990	-4.175542	5.197540	-6.656448	7.678446

Code for Analyis

```
#libraries used
library(geoR)
library(ggplot2)
library(knitr)
library(forecast)
library(lubridate)
library(tidyverse)
library(reshape)
library(patchwork)
#daily maximum temperatures data
temp<-read.csv("MaxTemp.csv")
#locations and elevation data
meta<-read.csv("metadata.csv")

#converting date in temp data set to date format
temp$Date=ymd(temp$Date)
#UK data for map
UK <- map_data(map = "world", region = "UK")
# melting september 12th data to get 1 row for every location
sept12<-melt(temp[256,],id.vars = "Date",variable_name ="Location",value.name="Maximum Temp
erature")
#merging september 12th with location to get coordinates
sept12<-merge(sept12,meta,on="Location")

#UK map with  september 12th temperatueres and elevations
a<-ggplot() +
  geom_polygon(data = UK, aes(x = long, y = lat, group = group),
```

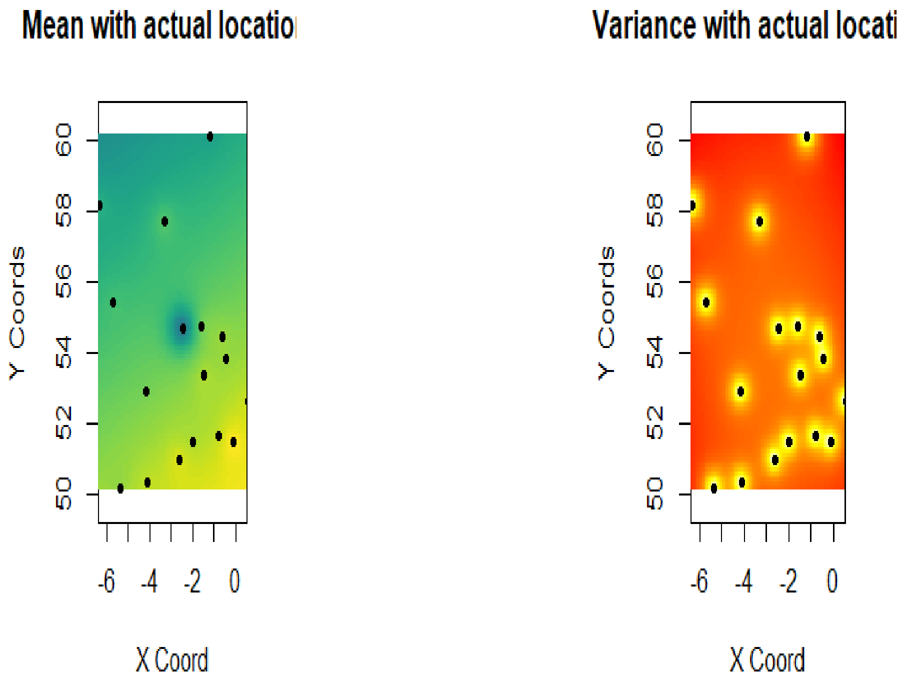
```
                         fill = 'gray90', color = 'black') +
    geom_point(data=sept12,aes(x=Longitude,y=Latitude,colour=value,size=Elevation))+
    coord_fixed(ratio = 1.3, xlim = c(-10,3),ylim = c(50, 59))+
  labs(x="Longitude",y="Latitude",title="(a) Twenty location points on UK map")

#melting temp data
meltData <- melt(temp,id.vars = "Date")

#boxplot for all locations for outliers
ggplot(meltData, aes(factor(variable), value))+
 geom_boxplot() + facet_wrap(~variable, scale="free")

#removing morecambe, coventry and kinross for september 12 th data
predicts<-sept12[c(2,6,14),]

sept12<-sept12[-c(2,6,14),]

#converting sept12 data to geodata
set.seed(356)
#for sept 12
gdata1 <- as.geodata(sept12,coords.col=4:5,data.col=3) # selecting maximum temperatues
dup <- dup.coords(gdata1)
gdata1 <- jitterDupCoords(gdata1,max=0.1,min=0.05) # jittering duplicates (geoR can't handle differ
ent data values in the same position)

#similary for prediction locations
gdata3 <- as.geodata(predicts,coords.col=4:5,data.col=3)
dup <- dup.coords(gdata3)
gdata3 <- jitterDupCoords(gdata3,max=0.1,min=0.05) # jittering duplicates (geoR can't handle differ
ent data values in the same position)

#plotting sept12 geo data
plot(gdata1)
```

```
#checking for isotropy
plot(variog4(gdata1))

#as we have seen spatial dependency including trend
plot(variog4(gdata1, trend = "1st"))

#For 2nd trend
plot(variog4(gdata1, trend = "2nd"))

#plotting variogram
plot(variog(gdata1, trend = "1st"))

## variog: computing omnidirectional variogram
```

```
#from variogram we take initial values
inits<-c(4,0.5)
#matern model for data with 1st trend maximum likelihood approach
ml_matern<-likfit(gdata1,cov.model = "matern",kappa = 1.5,trend="1st",ini=inits,fix.nugget = F,
                 fix.kappa = T)
```

```
#matern model for data with 1st trend Restricted maximum likelihood approach
reml_matern<-likfit(gdata1,cov.model = "matern",kappa = 1.5,trend="1st",ini=inits,
                    fix.nugget = F,fix.kappa = T,
                    lik.method = 'REML')

#cross validation for reml method
xv.matern<-xvalid(gdata1,model = reml_matern)

## xvalid: number of data locations       = 17
## xvalid: number of validation locations = 17
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17,
## xvalid: end of cross-validation

par(mfcol=c(5,2),mar=c(4,4,1,1))
plot(xv.matern)
```

```
#grid of coordinates in sampled data on 0.5 degree
pred.grid <- expand.grid(seq(min(gdata1$coords[,1]),max(gdata1$coords[,1]),by=0.1),
                         seq(min(gdata1$coords[,2]),max(gdata1$coords[,2]),by=0.1))
#predictions for matern model 1st trend
kc <- krige.conv(gdata1, loc = pred.grid, krige = krige.control(obj.model = reml_matern))

## krige.conv: model with mean given by a 1st order polynomial on the coordinates
## krige.conv: Kriging performed using global neighbourhood

par(mfrow=c(1,2))
#Image for mean of predictions with actual location for reml_matern1 model
image(kc, col = viridis::viridis(100), zlim = c(0,max(c(kc$predict))),
      coords.data = gdata1[1]$coords, main = 'Mean with actual locations',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))

#Image for variance of predictions with actual location for reml_matern1 model
image(kc, values = kc$krige.var, col =  heat.colors(100)[100:1],
      zlim = c(0,max(c(kc$krige.var))), coords.data = gdata1[1]$coords,
      main = 'Variance with actual locations',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))
```




```
#Image for mean of predictions with prediction location for reml_matern1 model
image(kc, col = viridis::viridis(100), zlim = c(0,max(c(kc$predict))),
    coords.data = gdata3[1]$coords, main = 'Mean with prediction locations',
    xlab = 'X Coord', ylab = 'Y Coords',
    x.leg = c(1,4), y.leg = c(51, 51.5))


#Image for mean of predictions with prediction location for reml_matern1 model
image(kc, values = kc$krige.var, col =  heat.colors(100)[100:1],
    zlim = c(0,max(c(kc$krige.var))), coords.data = gdata3[1]$coords,
    main = 'Variance with predicted locations',
    xlab = 'X Coord', ylab = 'Y Coords',
    x.leg = c(1,4), y.leg = c(51, 51.5))



#exponential method for ML approach
ml_exponential<-likfit(gdata1,cov.model = "exponential",ini=inits,trend="1st",fix.nugget = F)

#exponential for REML approach
reml_exponential<-likfit(gdata1,cov.model = "exponential",trend="1st",ini=inits,fix.nugget = F,
                lik.method = 'REML')

#cross validation for exponential method
xv.exponential<-xvalid(gdata1,model = reml_exponential)

## xvalid: number of data locations       = 17
## xvalid: number of validation locations = 17
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
```

```
17,
## xvalid: end of cross-validation

par(mfcol=c(5,2),mar=c(4,4,1,1))
plot(xv.exponential)
```

```
#prediction for reml exponential model
kc_exp <- krige.conv(gdata1, loc = pred.grid, krige = krige.control(obj.model = reml_exponential))

## krige.conv: model with mean given by a 1st order polynomial on the coordinates
## krige.conv: Kriging performed using global neighbourhood

par(mfrow=c(1,2))
#Image for mean of predictions with reml_exponential model
image(kc_exp, col = viridis::viridis(100), zlim = c(0,max(c(kc_exp$predict))),
      coords.data = gdata3[1]$coords, main = 'Mean-exponential',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))

#Image for variance of predictions with reml_exponential model
image(kc_exp, values = kc_exp$krige.var, col = heat.colors(100)[100:1],
      zlim = c(0,max(c(kc_exp$krige.var))), coords.data = gdata3[1]$coords,
      main = 'Variance-Matern REML 1st trend',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))
```

```
#for 26th april
aprl26<-melt(temp[117,],id.vars = "Date",variable_name ="Location",value.name="Maximum Tempe
rature")
aprl26<-merge(aprl26,meta,on="Location")

#converting april 26 data to geo data
gdata4 <- as.geodata(aprl26,coords.col=4:5,data.col=3) # selecting max temp
dup <- dup.coords(gdata4)
gdata4 <- jitterDupCoords(gdata4,max=0.1,min=0.05) # jittering duplicates (geoR can't handle differ
ent data values in the same position)

#REML matern covariane method
reml_matern_aprl25<-likfit(gdata4,cov.model = 'matern',kappa = 1.5,ini=inits,
                          trend = "1st",fix.nugget = F)

## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```r
#cross validation for matern method for april 26
xv.matern_a<-xvalid(gdata1,model = reml_matern_aprl25)
```

```
## xvalid: number of data locations       = 17
## xvalid: number of validation locations = 17
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 
17,
## xvalid: end of cross-validation
```

```r
par(mfcol=c(5,2),mar=c(4,4,1,1))
plot(xv.matern_a)
```

```r
#prediction grid for april 26th
pred.grid1 <- expand.grid(seq(min(gdata4$coords[,1]),max(gdata4$coords[,1]),by=0.1),
                          seq(min(gdata4$coords[,2]),max(gdata4$coords[,2]),by=0.1))
#predictions for matern model 1st trend for apil 25
kc_a <- krige.conv(gdata4, loc = pred.grid1, krige = krige.control(obj.model = reml_matern_aprl25))

par(mfrow=c(1,1))
#Image for mean of predictions with reml_matern model for april 26
image(kc_a, col = viridis::viridis(100), zlim = c(0,max(c(kc_a$predict))),
      coords.data = gdata4[1]$coords, main = 'Mean for April 26',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))


#Image for variance of predictions with reml_matern1 model for april 26
image(kc_a, values = kc_a$krige.var, col =  heat.colors(100)[100:1],
      zlim = c(0,max(c(kc_a$krige.var))), coords.data = gdata4[1]$coords,
      main = 'Variance for April 26',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))

#for dec 31st
dec31<-melt(temp[366,],id.vars = "Date",variable_name ="Location",value.name="Maximum Tempe
rature")
dec31<-merge(dec31,meta,on="Location")

#converting dec31 data to geodata
gdata5 <- as.geodata(dec31,coords.col=4:5,data.col=3) # selecting max temp
dup <- dup.coords(gdata5)
gdata5 <- jitterDupCoords(gdata5,max=0.1,min=0.05) # jittering duplicates (geoR can't handle differ
ent data values in the same position)

#REML matern covariate model for 1st trend
reml_matern_dec31<-likfit(gdata5,cov.model = 'matern',kappa = 1.5,ini=inits,
                          trend = "1st",fix.nugget = F)
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
```

```
##        times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```
#cross validation for matern method foe dec 31
xv.matern_d<-xvalid(gdata1,model = reml_matern_dec31)
```

```
## xvalid: number of data locations       = 17
## xvalid: number of validation locations = 17
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17,
## xvalid: end of cross-validation
```

```
par(mfcol=c(5,2),mar=c(4,4,1,1))
plot(xv.matern_d)
```

```
#prediction grid for dec 31
pred.grid2 <- expand.grid(seq(min(gdata5$coords[,1]),max(gdata5$coords[,1]),by=0.1),
                          seq(min(gdata5$coords[,2]),max(gdata5$coords[,2]),by=0.1))
#predictions for matern model 1st trend for dec 31
kc_d <- krige.conv(gdata5, loc = pred.grid2,
                   krige = krige.control(obj.model = reml_matern_dec31))


par(mfrow=c(1,1))
#Image for mean of predictions with reml_matern1 model for dec 31
image(kc_d, col = viridis::viridis(100), zlim = c(-2,max(c(kc_d$predict))),
      coords.data = gdata5[1]$coords, main = 'Mean for Dec 31',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))


#Image for variance of predictions with reml_matern1 model
image(kc_d, values = kc_d$krige.var, col =  heat.colors(100)[100:1],
      zlim = c(0,max(c(kc_d$krige.var))), coords.data = gdata5[1]$coords,
      main = 'Variance for Dec 31',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))

#for july 5th
jul5<-melt(temp[187,],id.vars = "Date",variable_name ="Location",
             value.name="Maximum Temperature")
jul5<-merge(jul5,meta,on="Location")

#converting july 5 data to geo data
gdata6 <- as.geodata(jul5,coords.col=4:5,data.col=3) # selecting max temp
dup <- dup.coords(gdata6)
gdata6 <- jitterDupCoords(gdata6,max=0.1,min=0.05) # jittering duplicates (geoR can't handle differ
ent data values in the same position)

#reml matern covariance model with 1st trend for july 5
reml_matern_jul5<-likfit(gdata4,cov.model = 'matern',kappa = 1.5,ini=inits,
                         trend = "1st",fix.nugget = F)
```

```
## ----------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ----------------------------------------------------------------
## likfit: end of numerical maximisation.

#cross validation for matern method for july 5
xv.matern_j<-xvalid(gdata1,model = reml_matern_jul5)

## xvalid: number of data locations       = 17
## xvalid: number of validation locations = 17
## xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17,
## xvalid: end of cross-validation

par(mfcol=c(5,2),mar=c(4,4,1,1))
plot(xv.matern_j)
```

```
#prediction grid for july 5
pred.grid3 <- expand.grid(seq(min(gdata6$coords[,1]),max(gdata6$coords[,1]),by=0.1),
                          seq(min(gdata6$coords[,2]),max(gdata6$coords[,2]),by=0.1))
#predictions for matern model 1st trend
kc_ju <- krige.conv(gdata6, loc = pred.grid3,
                    krige = krige.control(obj.model = reml_matern_jul5))

## krige.conv: model with mean given by a 1st order polynomial on the coordinates
## krige.conv: Kriging performed using global neighbourhood

par(mfrow=c(1,1))
#Image for mean of predictions with reml_matern1 model
image(kc_ju, col = viridis::viridis(100), zlim = c(0,max(c(kc_ju$predict))),
      coords.data = gdata6[1]$coords, main = 'Mean for July 5',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))
```

```
#Image for variance of predictions with reml_matern1 model
image(kc_ju, values = kc_ju$krige.var, col =  heat.colors(100)[100:1],
      zlim = c(0,max(c(kc_ju$krige.var))), coords.data = gdata6[1]$coords,
      main = 'Variance for July 5',
      xlab = 'X Coord', ylab = 'Y Coords',
      x.leg = c(1,4), y.leg = c(51, 51.5))
```

Time series Analysis

```
#converting temp data to time series
time_series<-ts(temp,start = 1,frequency = 1)
```

```
#time series data for yeovilton, morecambe, sheffield, london, lerwick, camborne, kinross,lossiemouth
yev<-time_series[,"Yeovilton"]
mor<-time_series[,"Morecambe"]
sheff<-time_series[,"Sheffield"]
lon<-time_series[,"London"]
hig<-time_series[,"High_Wycombe"]
ler<-time_series[,"Lerwick"]
cam<-time_series[,"Camborne"]
loss<-time_series[,"Lossiemouth"]
kin<-time_series[,"Kinross"]

#storing date of temp data
Date<-temp$Date

#time series plot for all the above locations
b<-ggplot()+geom_line(aes(x=Date,y=yev))+
  geom_line(aes(x=Date,y=mor,colour="Morecambe"))+
  geom_line(aes(x=Date,y=sheff,colour="Sheffield"))+
  geom_line(aes(x=Date,y=lon,colour="London"))+
  geom_line(aes(x=Date,y=hig,colour="High Wycombe"))+
  geom_line(aes(x=Date,y=ler,colour="Lerwick"))+
  geom_line(aes(x=Date,y=cam,colour="Camborne"))+
  geom_line(aes(x=Date,y=loss,colour="Lossiemouth"))+
  geom_line(aes(x=Date,y=kin,colour="Kinross"))+
  labs(y="Maximum Temperatures",x="Date",
       title = "(b) Maximum temperatures in a day of some locations in 2020")

#date till october 31st
Date1<- temp$Date[1:305]
#yeovilton data till october 31
yev1<-time_series[,"Yeovilton"][1:305]
#yeovilton time series till october
c<-ggplot()+geom_line(aes(x=Date1,y=yev1),colour="blue")+
  labs(y="maximum temperatue",title="a) Yeovilton time series till October")


#function to check number of differencing required

ndiff(yev1)

## 1


#difference
d<-ggplot()+geom_line(aes(x=Date1[1:304],y=diff(yev1)),colour="blue")+
  labs(x="Date",y="Difference",title="b) 1st diffrencing of Yeovilton time series")

par(mfrow=c(1,2))
acf(diff(yev1),main="ACF");pacf(diff(yev1),main="PACF")
```

```
#auto arima model for
model1<-auto.arima(yev1, max.p = 4, max.q = 4, max.d = 4, seasonal = FALSE)
model1
```

```
## Series: yev1
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.7203  -0.1199  -0.9085
## s.e.  0.0641   0.0601   0.0326
##
## sigma^2 = 5.998:  log likelihood = -702.48
## AIC=1412.96   AICc=1413.09   BIC=1427.83
```

```
#model with significant co-efficients
model2<-arima(yev1,order=c(1,1,2))
model2
```

```
##
## Call:
## arima(x = yev1, order = c(1, 1, 2))
##
## Coefficients:
##          ar1      ma1      ma2
##       0.5393  -0.7237  -0.1716
## s.e.  0.0935   0.1020   0.0796
##
## sigma^2 estimated as 5.933:  log likelihood = -702.35,  aic = 1412.7
```

```
tsdiag(model2)
```

```
#prediction from 1st n0vember to 7th november
yev_preds<-forecast(model2,h=7)

#for model appropriateness
#london data till october 31st
lon1<-time_series[,"London"][1:305]
#sheffield data till october 31st
shef1<-time_series[,"Sheffield"][1:305]

#model for london
model3<-arima(lon1,order = c(1,1,2))
model3
```

```
##
## Call:
## arima(x = lon1, order = c(1, 1, 2))
##
## Coefficients:
##          ar1      ma1      ma2
##       0.6214  -0.7363  -0.1647
## s.e.  0.0857   0.0944   0.0707
##
## sigma^2 estimated as 6.64:  log likelihood = -719.4,  aic = 1446.8
```

```
#model for sheffield
model4<-arima(shef1,order = c(1,1,2))
model4

##
## Call:
## arima(x = shef1, order = c(1, 1, 2))
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4550  -0.6749  -0.2141
## s.e.  0.1021   0.1068   0.0833
##
## sigma^2 estimated as 8.222:  log likelihood = -751.99,  aic = 1511.98

#residuals for London model
tsdiag(model3)

#residuals for Sheffield model

tsdiag(model4)
```

```
 #london prediction from 1st to 20 november
lon1_preds<-forecast(model3,h=20)

#table for predictions along with CI and actual values
lon_data_table<- data.frame(Date=Date[306:325],lon1_preds,actual=lon[306:325])
kable(lon_data_table)

#sheffield prediction from 1st to 20 november
shef1_preds<-forecast(model4,h=20)

#table for predictions along with CI and actual values
shef_data_table<- data.frame(Date=Date[306:325],shef1_preds,actual=sheff[306:325])
kable(shef_data_table)

#london and Dun fell data for daily analysis
lon<-time_series[,"London"]
dun<-time_series[,"Dun_Fell"]

#london time serie plot
e<-ggplot()+geom_line(aes(x=Date,y=lon),colour="blue")+
  labs(y="maximum temperatue",title="a) London time series data")

#dun fell time series plot
f<-ggplot()+geom_line(aes(x=Date,y=dun),colour="blue")+
  labs(y="maximum temperatue",title="b) Dun fell time series data")

#as model arima(1,1,2) was appropriate
#building same order model
#for london
model5<-arima(lon,order=c(1,1,2))
model5
```

```
##
## Call:
## arima(x = lon, order = c(1, 1, 2))
##
## Coefficients:
##          ar1      ma1      ma2
##       0.5929  -0.7189  -0.1679
## s.e.  0.0891   0.0983   0.0707
##
## sigma^2 estimated as 6.556:  log likelihood = -861.36,  aic = 1730.72
```

```
#for dun fell
model6<- arima(dun,order=c(1,1,2))
model6
```

```
##
## Call:
## arima(x = dun, order = c(1, 1, 2))
##
## Coefficients:
##          ar1      ma1      ma2
##       0.4337  -0.6375  -0.2520
## s.e.  0.0862   0.0863   0.0676
##
## sigma^2 estimated as 6.821:  log likelihood = -868.75,  aic = 1745.49
```

```
#residual plots for both models
tsdiag(model5)
```

```
tsdiag(model6)
```

```
#prediction of london from 1st january 2021 to 14th january 2021
lon_daily_preds<- forecast(model5, h=14)
#prediction of dun fell from 1st january 2021 to 14th january 2021
dun_daily_preds<- forecast(model6, h=14)

#dates from 1st january 2021 to 14th january 2021
Date_ext<-as.Date('2021-01-01')+ 0:13
#table for predictions and CI for london
lon_daily_table<-data.frame(Date=Date_ext,lon_daily_preds)
kable(lon_daily_preds)
```

```
#table for predictions and CI for Dun fell
dun_daily_table<-data.frame(Date=Date_ext,dun_daily_preds)
kable(dun_daily_preds)
```

```
#weekly analysis
#rounding dates to weeks
temp$week <- floor_date(temp$Date, "week")

#find mean of locations  by week
weekly_data<-temp %>%
            group_by(week) %>%
            summarize_at(.cols = 2:21,.funs = mean,na.rm=T)
```

```r
#converting weekly data to time series
weekly_time<-ts(weekly_data[,-1])

#weekly data for london
london_week<- weekly_time[,"London"]
#weekly data for dun fell
dunfell_week<-weekly_time[,"Dun_Fell"]

#plot for weekly data for london
g<-ggplot()+geom_line(aes(x=time(weekly_time),y=london_week))+
  labs(x="Week",y="Average maximum temperature weekly",
       title = "Time series of weekly average temperatures for london 2020")
#plot for weekly data for dun fell
h<-ggplot()+geom_line(aes(x=time(weekly_time),y=dunfell_week))+
  labs(x="Week",y="Average maximum temperature weekly",
       title = "Time series of weekly average temperatures for Dun fell 2020")

par(mfrow=c(1,3))
#acf and pacf for london
plot(diff(london_week), main="Differnce of london weekly data");
acf(diff(london_week),main="ACF");pacf(diff(london_week),main="PACF")
```

```r
#acf and pacf for dun fell differnc data
plot(diff(dunfell_week), main="Difference of Dun fell weekly data");
acf(diff(dunfell_week),main="ACF");pacf(diff(dunfell_week),main="PACF")
```

```r
#auto arima for london weekly data
model7<-auto.arima(london_week,max.p = 4,max.q = 4,max.d = 4,seasonal = TRUE)
model7

## Series: london_week
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.1515  -0.5863  -1.6218  0.9486
## s.e.  0.1606   0.1457   0.1513  0.1561
##
## sigma^2 = 8.625:  log likelihood = -128.96
## AIC=267.93   AICc=269.23   BIC=277.68

#residuals for model7
tsdiag(model7)
```

```r
#arima model for dunfell arima(2,1,2)
model8<-arima(dunfell_week,order=c(2,1,2))

#residuals for model8
tsdiag(model8)
```

```r
Week<-54:57
#prediction for next 4 weeks for london
london_week_preds<-forecast(model7, h=4)
#table with predictions and CI
kable(london_week_preds)

#plot with predictions and CI
i<-ggplot()+
  geom_line(aes(x=Week,y=london_week_preds$mean,colour="Predict"))+
  geom_line(aes(x=Week,y=london_week_preds$lower[,2],colour="Lower CI"))+
  geom_line(aes(x=Week,y=london_week_preds$upper[,2],colour="Upper CI"))+
  labs(y="Average maximum temperature weekly",
       title = "Prediction of weekly maximum temperatures of london")

#prediction for next 4 weeks for dunfell
dunfell_week_preds<-forecast(model8, h=4)
#table with predictions and CI
kable(dunfell_week_preds)

#plot with predictions and CI
j<-ggplot()+
  geom_line(aes(x=Week,y=dunfell_week_preds$mean,colour="Predict"))+
  geom_line(aes(x=Week,y=dunfell_week_preds$lower[,2],colour="Lower CI"))+
  geom_line(aes(x=Week,y=dunfell_week_preds$upper[,2],colour="Upper CI"))+
  labs(y="Average maximum temperature weekly",
       title = "Prediction of weekly maximum temperatures of dunfell")
```