A Spatio-Temporal Analysis of UK Rainfall Data

1. Introduction and Data Description

1.1 Aims

The aim of the project is to analyse the spatio-temporal characteristics of historical UK rainfall data and to employ statistical modelling techniques, such as Autoregressive Integrated Moving Average (ARIMA) and Artificial Neural Networks (ANN), to determine whether the data shows attributes that enables it to be modelled based on its past values such that future values can be accurately forecast.

Time series analysis involves designing a model to enable extrapolation into the future based on the expectation that historic patterns will repeat over time. Several time series modelling methodologies have been developed of which ARIMA is one. ARIMA models have their roots in electrical engineering [1] and were first adapted for analysis of time series using statistical methods by Box and Jenkins in the 1970s [2]. However, there are limitations with ARIMA models, some of which will be discussed including the stationarity requirement and ability to deal with non-linear data. This has led to the emergence of artificial intelligence models such as ANN as powerful alternatives for time series forecasting [3]. However, there are conflicting opinions on the ability of ANNs to forecast seasonal timeseries. Authors such as Benkachacha, et al (2015)[4] have found ANN models to come with lower prediction errors than other methods (such as ARIMA), whereas Zhang & Qi (2005)[5] argue ANNs yield mixed results and are not able to effectively capture seasonal or trend variations.

1.2 UK Regional Rainfall Data

Monthly rainfall records for the UK were sourced from the UK Met Office [6]. The inspiration for using this data came from an article describing how the data was recently updated through the citizen science project 'Rainfall Rescue' [7]. The data covers the period January 1836 to December 2021 (2232 months) for each of the 10 district regions of the UK as defined by the UK Met Office [8].

Data for each region was downloaded as a .txt file, combined and indexed in an Excel file, and converted to .csv format (“uk_rainfall_data.csv”) to enable them to be joined with shapefiles. Shapefiles for the Great Britain district regions were sourced from the CEGE0042 Tutorial data. A further shapefile for Northern Ireland was sourced from OSNI Open Data [9]. These were combined into a single shapefile and then spatially joined with the rainfall data in ArcGIS Pro to create a geodatabase feature layer for importing as a dataframe into R: layer name “uk_rain_all_districts” in file “UK Rainfall.gdb”.

1.3 UK Weather Station Rainfall Data

Monthly UK rainfall data for weather stations sites across the UK was also sourced from the UK Met Office to give a point dataset to complement the areal dataset described in 1.2. The complete record contains monthly data for 37 weather station sites, with records of differing lengths going back as far as 1853 in some cases (see “Station Data.xlsx”). Due to not all the weather station sites having complete records, a subset of sites was identified that had near complete records for a significant period (Sep’64 to Aug’16 (445 months)) and the corresponding rainfall data was segregated for analysis (see worksheet “Selected Combined” in file “Station Data.xlsx”). Gaps in the data for individual months for individual stations were filled using prior year ratio values to create continuous record and to avoid the issue of NAs when working with the data in R (highlighted yellow in worksheet “rainfall_by_station” in file “Station Data.xlsx”).

Histograms of both the monthly regional data and the monthly weather station point data show non-normal distribution of rainfall values, bounded by zero with a positive skew due to small numbers of months with very high rainfall (Figure 1). Mean monthly rainfall for all regions for the full period of 1836-2021 is 88.55mm, with a standard deviation of 51.33mm. The ‘Scotland W’ region has the highest mean monthly rainfall at 131.78mm, with ‘East Anglia’ the lowest at 51.12mm.

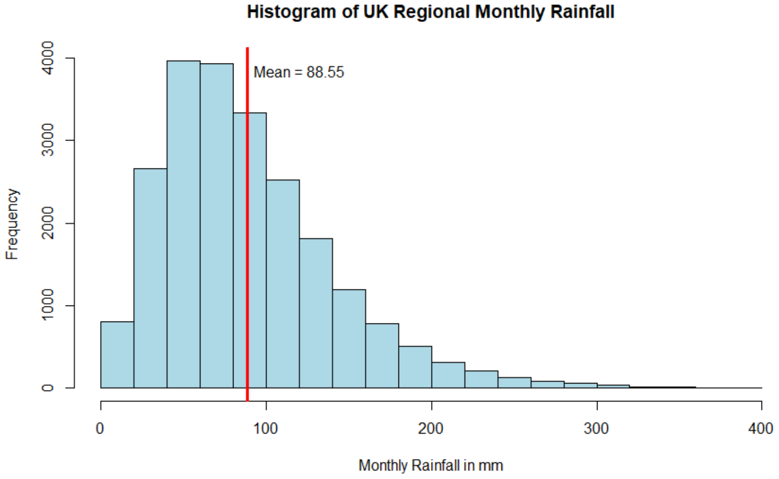

Figure 1: Histogram of monthly rainfall data for UK regions

2. Exploratory Spatio-Temporal Data Analysis

2.1 Spatial and Temporal Characteristics

Summary visualisation of the full UK Regional Rainfall dataset is challenging due to the sheer

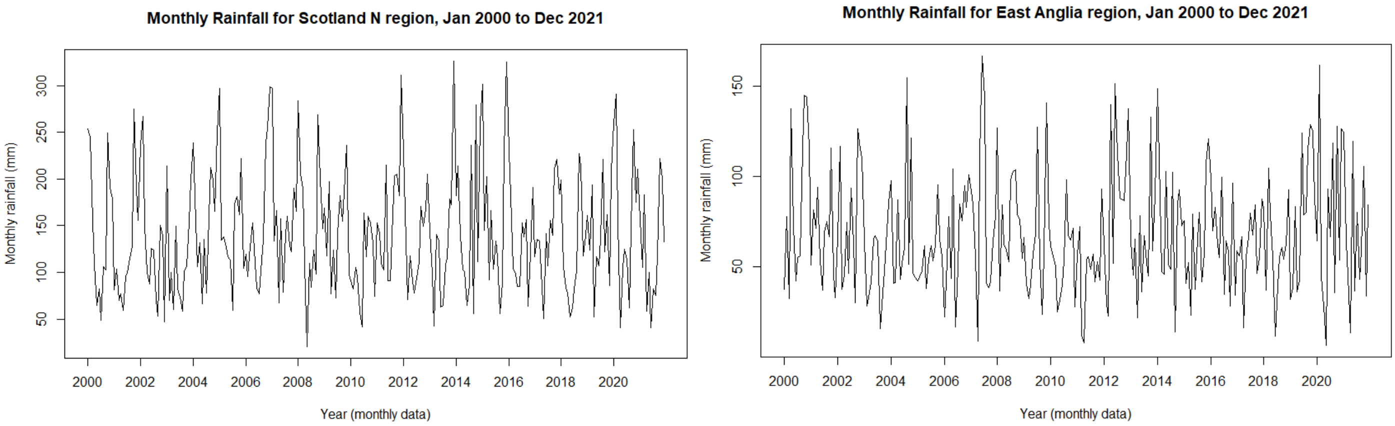

Figure 2: Plots of monthly rainfall time series for Scotland N and East Anglia regions for Jan 2000 to Dec 2021

number of months and due to there being 10 regions. A sample of the data for a shorter period for two of the regions was chosen and plotted as a time series (Figure 2). The data shows considerable variation from month to month, little evidence of a distinct long-term trend, which is consistent with the findings of Lee (2020) [10].

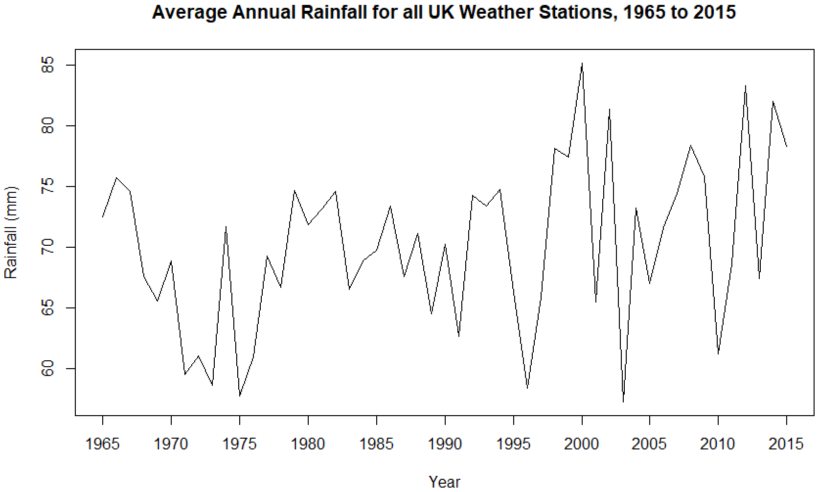

Figure 3: Average annual rainfall for all UK weather stations, 1965 to 2015

Plotting the monthly UK weather station data reveals a similar pattern to the regional data, with significant variation from month-to-month, no obvious long-term trend up or down, and suggestions of seasonality. Plotting the annual averages showed no obvious patterns (Figure 3), indicating there has been no overall change in UK annual rainfall levels on the past 50-60 years, which is consistent with the findings of Jenkins, et al. (2009) [11].

2D, 3D and dynamic scatterplots of the UK weather station point data do not show any clear relationships between average rainfall levels and latitude, longitude, or altitude, other than a hint that rainfall is higher the further west and north. This relationship is reinforced by examining heatmaps ordered by latitude and longitude, which confirm highest rainfall levels in the north and west. Latitude appears to be a greater factor than longitude. This could be due to the shape of the UK, which has a greater north-south extent than east-west. These heatmaps also show that the spatial variation in rainfall is greater than the temporal, as most of the rows are consistent in colour.

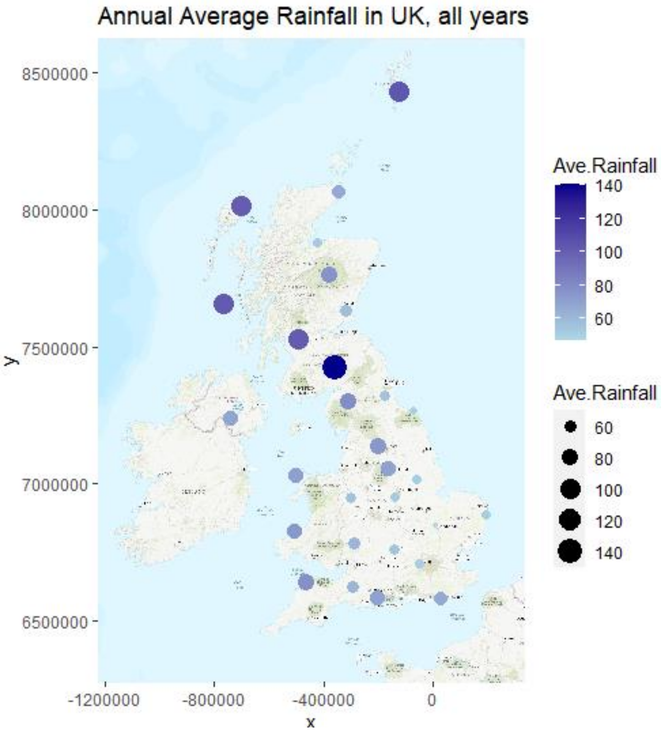

Figure 4: Annual average rainfall for UK weather stations, 1965 to 2015

Mapping of the annual average rainfall figures for the weather station point data also confirmed the pattern of wetter in the north and west, drier in the south and east and away from the coasts (Figure 4). Eskdalemuir in Scotland records the highest annual average rainfall (140mm); Cambridge records the lowest (46mm). This is in line with expectations based on knowledge of the south-westerly prevailing weather pattern in the UK, which brings rain in from the Atlantic with the central hills and mountains of England and Scotland creating a rain shadow effect to the east. Choropleths for each month of 1984, which was identified by analysis in Excel to be year with each region most closely matching their annual average rainfall, also clearly shows that the western and northern regions of the UK are the wettest, although 'N. Ireland' is relatively drier than nearby 'Scotland W'. The driest regions are towards the east and the south. Looking just at 1984, there is an indication of seasonality with more rain in the autumn/winter than spring/summer.

2.2 Spatial and Temporal Autocorrelation

Global Moran's I was calculated to test the spatial autocorrelation of rainfall levels across the regions of the UK. This had to be done for GB, i.e., excluding N. Ireland, which otherwise created an empty neighbour set as it is not connected to the rest of the UK. A Global Moran's I value of 0.516 was calculated within a range of -0.593 to 1.018. The moran.test and moran.mc functions both gave p-values <0.05, confirming statistically significant autocorrelation for the regional data, leading to the conclusion that rainfall in one region is more similar in neighbouring regions than those further away.

The lowest local spatial autocorrelation values are in the 'S Wales & England SW' and 'England NW & N Wales' regions. These are wet regions in the west with long borders with dry regions in the east, hence low autocorrelation. The highest local Moran's I values are seen in 'East Anglia', a dry region in the east bordering other dry regions, and 'Scotland N', a wet region in the north-west bordering other wet regions. Based on unadjusted p-values, only 'Scotland W' and 'East Anglia' have significant local Moran's I, but adjusting the p-values using the Bonferroni method shows no regions with statistically significant local Moran's I.

Spatial autocorrelation in the UK weather station point data was analysed using a semivariogram. Results show a scattered result but there is an indication that rainfall levels at weather stations that are closer are more similar than those further away. No clear results were seen from the directional variograms, although there are hints of anisotropy in that not all the semivariograms look the same, with the semivariance varying more with distance in the 0^{o} and 135^{o} plots than the 45^{o} and 90^{o} plots. This gives a weak indication that spatial autocorrelation is stronger in the north-south and northwest-southeast directions than in other directions, which is broadly consistent with the findings from the analysis of spatial characteristics.

Temporal autocorrelation within the regional dataset shows it to be generally weak, with one month's rainfall less strongly correlated to previous month's rainfall than seen with UK temperatures. The 'Scotland N' region shows the highest temporal autocorrelation with a PMCC of 0.306; 'Midlands' shows lowest PMCC of 0.081 (1 month lag interval).

The annual data for UK weather stations also shows week temporal autocorrelation (PMCC = 0.121), indicating one year's rainfall is not significantly related to the previous year's rainfall.

3. Methodology and Results

3.1 ARIMA

For the analysis of ARIMA (and ANN in section 3.2), this report now focuses solely on the UK regional rainfall data and the 'Scotland N' regional data in particular as this showed the strongest temporal dependency. A shortened version of the 'Scotland N' data covering the period 1990 to 2021 is used to make the analysis more manageable. This time series is decomposed into its trend, seasonal and residuals components to determine stationarity and then the Box-Jenkins approach to ARIMA modelling is followed to try to fit and test a model for forecasting.

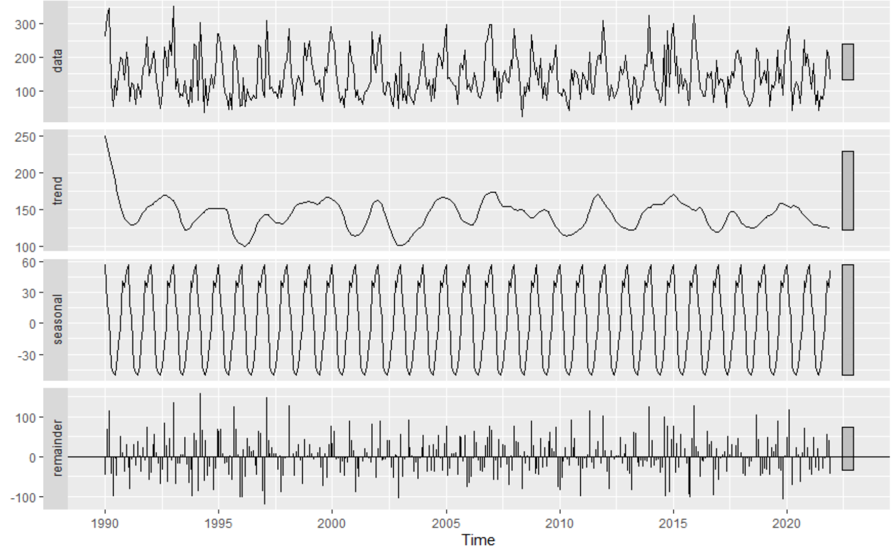

Figure 5: Decomposition of Scotland N regional rainfall data 1990 to 2021 using STL

It is expected that the 'Scotland N' time series is not stationary due to it containing an element of seasonality. Decomposition of the 'Scotland N' time series was done using Seasonal and Trend Decomposition using Loess (STL). Various values for the t.window parameter were tried, with a value of 25 chosen as this resulted in the smallest remainders. The results (Figure 5) show a trend component with no clear pattern, but a clear seasonal component. However, remainders are still high at approximately +/-100mm, compared to the seasonal component, which is +/-50 to 60mm. Confirmation of a clear seasonal component indicates that seasonal differencing will be required for ARIMA. With the first step – Exploratory Data Analysis – already completed, the Box-Jenkins approach moves directly to the next step with analysis of the autocorrelation factor (ACF) and partial autocorrelation factor (PACF) plots and differencing to provide more insight into potential ARIMA model parameters. The ACF plot of the 'Scotland N' regional data (Figure 6, top) shows a seasonal pattern with positive peaks at lags of 12, 24, and 36 months, and negative peaks at 6, 18, and 30 months. This is the hallmark of a non-stationary time series and strongly suggests seasonal differencing with order 12 is required to make the time series stationary, which is a prerequisite of a linear regression model

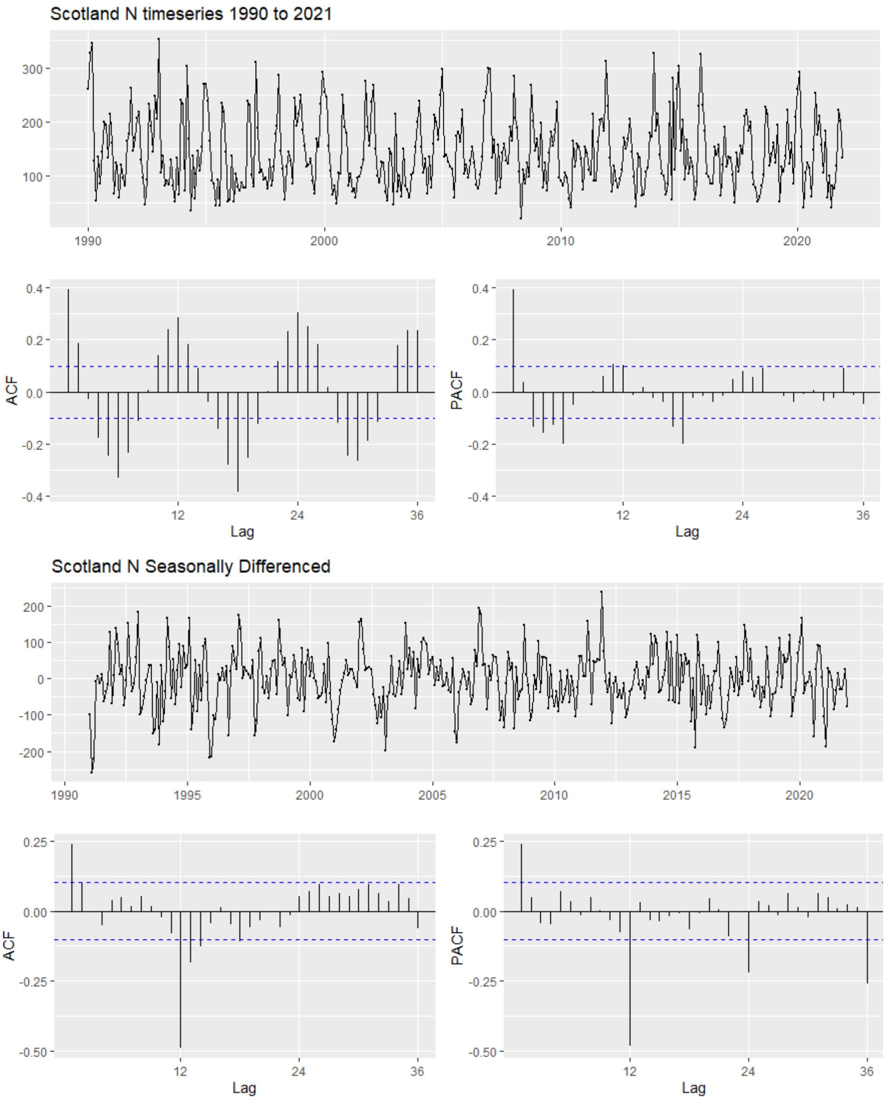

Figure 6: Scotland N timeseries 1990 to 2021 and associated ACF and PACF plots – Undifferenced (top), with Seasonal Differencing (bottom)

such as ARIMA (D=1, m=12). This was done and the ACF re-run, producing a plot showing that seasonal differencing has largely removed the seasonal component from the data (Figure 6, bottom) and the data now appears stationary. Significant ACF remain at lags 1 and 12 (seasonal lag 1), whilst the PACF shows a small but significant value at lag 1 and exponential decay in the seasonal lags. This suggests both seasonal and non-seasonal MA components maybe required (q=1, Q=1).

Based on the above, an initial model of ARIMA(0, 0, 1)(0, 1, 1)12 was selected for the parameter estimation and fitting part of the process. The Arima() function in R was used, as recommended by Hyndman and Athanasopoulos (2018) [12]. The model gave the following results: log likelihood = -11815.62, Akaike Information Criterion (AICc) = 23637.25. The residuals were checked for any remaining autocorrelations using the checkresiduals() function, which showed that the residuals look like white noise with no significant ACF spikes, except for one at lag 18, which suggests there is some remaining weak autocorrelation and that a better model could be found. The Ljung-Box test gives a p-value of 0.5426, which meets the condition of p >0.05 that suggests the model does not exhibit significant lack of fit and the residuals are white noise.

Alternative models were then tested to see if they produced better results based on the log likelihood (looking for a higher value) and AICc (looking for a lower value). No changes were made to the seasonal or non-seasonal differencing components of the models so that the AICc comparison remained valid. The results are summarised in Table 1.

Several of the models tested show very similar results. For example, ARIMA(1,0,0)(0,1,1)12 (fit7.Ar) gave very similar results to the initial model tested. The two models differ solely on having a non-seasonal AR component instead of a non-seasonal MA component. This suggests that there is little impact from the non-seasonal components of the model. Based solely on highest (least negative) log likelihood and lowest AICc value, the ARIMA(1,0,0)(2,1,1)12 model (fit6.Ar) performed the best, although this model still showed just about significant ACF value at lag 18, which suggests not all the autocorrelation has been removed. This model can still be used for forecasting but the correlated residuals may mean that the prediction intervals are not accurate [12]. Compared to the initial model, this model additionally has a non-seasonal AR component instead of an MA component, plus two seasonal AR components.

Model		Log Likelihood	AICc	NRMSE	p-Value (Ljung-Box)
Initial	ARIMA(0,0,1)(0,1,1)12	-11815.62	23637.25	0.84564	0.5426
fit2.Ar	ARIMA(1,0,1)(0,1,1)12	-11815.61	23639.23	0.84563	0.4288
fit3.Ar	ARIMA(0,0,1)(1,1,1)12	-11813.78	23635.58	0.84484	0.4112
fit4.Ar	ARIMA(0,0,1)(2,1,1)12	-11812.07	23634.17	0.84412	0.2925
fit5.Ar	ARIMA(1,0,1)(2,1,1)12	-11812.04	23636.11	0.84408	0.1938
fit6.Ar	ARIMA(1,0,0)(2,1,1)12	-11812.07	23634.16	0.84410	0.2733
fit7.Ar	ARIMA(1,0,0)(0,1,1)12	-11815.66	23637.34	0.84564	0.5173
fit8.Ar	ARIMA(1,0,0)(1,1,1)12	-11813.82	23635.66	0.84484	0.3826

Table 1: Summary of ARIMA model fit results

The best fitting model was used to create a training dataset on 176years of the full 186-year ‘Scotland N’ dataset with a view to predicting the remaining 10 years of the data and comparing to actuals. The Arima()

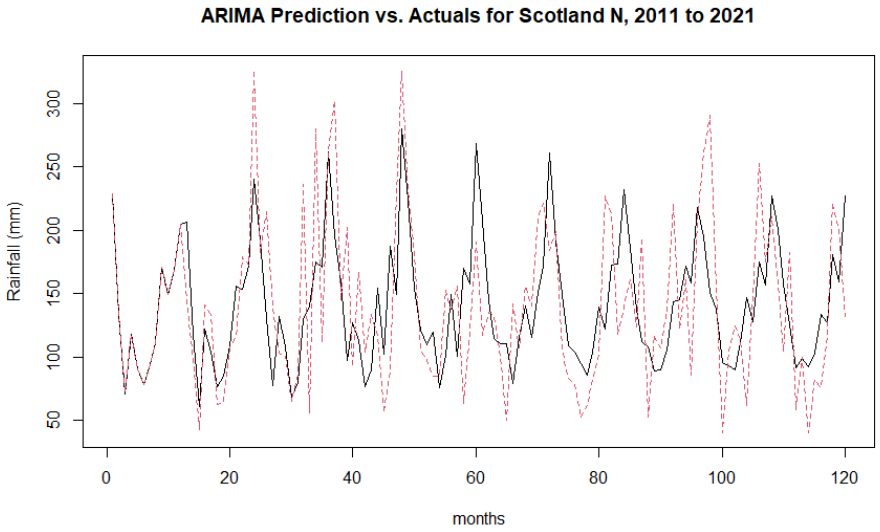

Figure 7: ARIMA model prediction (black) vs. actuals (red) for Scotland N region, 2011 to 2021

function uses a one-step ahead approach to forecasting such that the most recently available data is used at each point in time to make the next forecast. The results (Figure 7) show the prediction closely matches the actuals for the first 12-18 months, but then the model breaks down somewhat, particularly around months 30-40 where the model is smooth relative to the noisier actuals. The overall shape of the prediction is reasonably good but seems to have a long-term trend component to the amplitude of the seasonal highs and lows. Comparison was then made with the model found through the auto.arima() function. Running this function on the same Scotland N dataset did not result in a seasonal ARIMA model suggestion, which was somewhat surprising. Instead, an ARIMA(3,1,2) model was found. Whilst the AICc result cannot be directly compared with the results for the models tested per Table 1 due to the presence of a first difference component, analysis of the residuals showed significant ACF values at multiple lags that strongly suggested seasonality is not successfully accounted for in the auto.arima model, and the Ljung-Box p-values are much less than the 0.05 level, meaning this model does not pass the test for it to be considered a suitable model for the 'Scotland N' rainfall data. Using the auto.arima model for prediction of the last 10 years of the timeseries yields results that are arguably less accurate than for the model found manually.

3.2 Artificial Neural Networks

Neural Networks are a form of supervised learning where labelled training data used to predict labels for unseen data through adaptive discovery of patterns in the data. Neural networks have been shown to learn from experience and estimate complex functional relationships with high degrees of accuracy, given an appropriate number of non-linear processing units [5]. One type of ANN is the multilayer feed-forward network where one or mode hidden layers of nodes takes weighted inputs from an input layer and provides an output to an output layer consisting of one or more nodes via a non-linear function. This makes them suitable for modelling of non-linear (seasonal) data without the requirement seen in ARIMA of having stationary data before modelling and forecasting. This part of the project aims to test the ability of an ANN to forecast UK rainfall based on the historic rainfall data and to compare results with those of the ARIMA models discussed above.

A multi-input, multi-output ANN was constructed using the nnet() function in R, with the intention of using the lagged values of the timeseries to predict the rainfall for forward months. A training set of 80% of the full 186-year regional rainfall dataset for all ten regions was used to train the ANN. Unfortunately, the results achieved were not in line with expectations based on the analysis of UK temperature data provided in the class tutorial. Regardless of how the parameters of the nnet() function were adjusted, e.g., decay and size (number of nodes in hidden layer), it was not possible to generate a predicted dataset that consisted of anything other than a set of almost constant values for each region. Consequently, predicted values performed very poorly compared to actuals. It seems the data was converging to a mean value or something similar.

An alternative approach was taken using the nnetar() function, as described in Hyndman and Athanasopoulos (2018) [12]. This function automatically determines the parameters for the number of lags to use and the number of nodes in the hidden layer. If seasonality is identified in the training data, nnetar() will also add seasonal lags into the model. This is like the auto.arima() function discussed in 3.1.The results using nnetar() certainly looked more encouraging than for nnet(). Using the full 186-year dataset for the 'Scotland N' region, nnetar() generated a NNAR(29,15) model, meaning 29 monthly lags of data are input into a hidden layer containing 15 nodes. As with auto.arima, no seasonal component was added to the model. Using this model to generate a prediction of rainfall levels for a future 50 months looked reasonable when plotted, although there are no actuals for this period to use as a comparison to determine accuracy. Due to the length of the timeseries, it is hard to visualise effectively, so a shortened dataset was used containing the last 360 months (30 years) to forecast forward 50 months into the future. This produced a NNAR(18,10) model – 18 monthly lags, no seasonal component, 10 nodes in hidden layer – and the resulting forward prediction looks reasonable when plotted as an extension to the actuals, although the amplitude of seasonal variance looks dampened versus historic actuals (Figure 8). A method to create a prediction for a period that allowed direct comparison to, and plotting against, actuals was not found. So other than a visual inspection for reasonableness, no quantitative analysis of the NNAR model performance has been carried out.

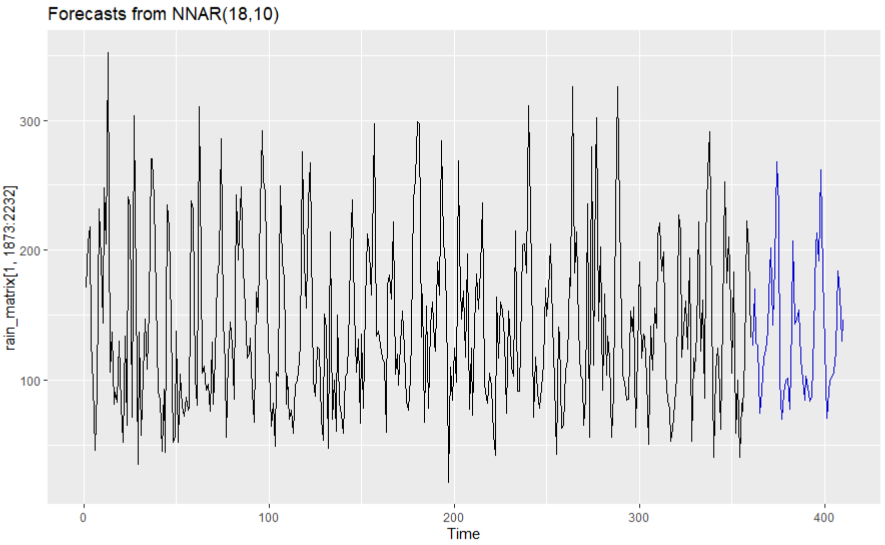

Figure 8: Actuals (black) and future prediction (blue) from nnetar() for Scotland N timeseries

4. Discussion & Conclusions

Inspired by the recent citizen science project "Rainfall Rescue" [6], historic rainfall data for the ten regions of the UK as defined by the UK Met Office were downloaded and pre-processed for analysis and visualisation by joining to shapefiles. Spatial and temporal analysis of the UK rainfall time series data shows it to be quite weakly correlated spatially and temporally. The wettest regions are in the north and west of the UK and there is some seasonality with the wettest months in the autumn and winter, but with no apparent long-term trend. Rainfall displays a seasonal characteristic, with more rain in autumn and winter than in spring and summer. However, this seasonality is not as clearly defined and consistent as seen with the UK temperature data analysed in the class tutorials.

ARIMA models were tested to determine their ability to forecast UK rainfall based on historic rainfall records. Decomposition of the timeseries showed a seasonal component to the data, but that it was not particularly strong. The process for determining appropriate AR and MA parameters for the ARIMA model is well defined in the literature (Box-Jenkins method), but the stationarity requirement and meant seasonal differencing was needed, which added complexity to the process of selecting a suitable ARIMA model based on the interpretation of ACF and PACF plots. Comparison of various candidate ARIMA models showed little difference between quantitative metrics when using the 'Scotland N' dataset (Table 1). The auto.arima() function in R came up with a model that was quite different from that derived manually that did not contain any seasonal components and which did not seem to make sense from inspection of the ACFs and PACFs. Based on the NRMSE metric, which is comparable across models with different orders of differencing, and on comparison of model diagnostics, the results of manual model for predicting rainfall appeared better than for auto.arima. Further analysis of this result is required to understand the outcome. The process of fitting the ARIMA models to the data and associated diagnostic checking did not appear to be computationally intensive. With a training dataset of 176 years, the ARIMA models were used to try and predict rainfall for the last 10 years. The manual model performed better than the auto.arima model in predicting 10 years of rainfall. Comparison with actuals for the manual model showed good results for the first 12-18 months, after which the model performed less well, although the overall seasonal shape of the prediction was reasonable. Possible explanations for this include sub-optimal AR and MA parameter choice leading to over or under fitting of the model to the data, or failure to identify other model parameters such as a constant or drift component, which could be explored as an extension of this project. Similar ARIMA models could be derived through analysis of other regions of the UK to test whether the results seen for 'Scotland N' are consistent.

Due to the non-linearity of the UK rainfall data, ANN models were also examined for comparison to ARIMA methods. In this project, the nnet() and nnetar() functions in R were used on the full regional UK rainfall timeseries consisting of 2232 monthly observations across 10 regions. Unfortunately, the results achieved using nnet() were far from satisfactory, regardless of the choice of decay and size parameter. Results appeared to converge to a mean monthly rainfall in the predicted values for each region and almost all seasonality was lost. The reasons for these results are not understood but are possibly due to insufficient lagged data being included in the model, leading to under-fitting of the model. Only one month lag was used, whereas, in comparison, the nnetar() function, which automatically derives the number of lags to use, used 29 monthly lags on the full 186-year dataset and 18 monthly lags on the 30-year training dataset. Using a one-month lag, the data was seen to be only weakly temporally autocorrelated, which may support this possibility. Using the nnetar() function to forecast out beyond the end of the timeseries appeared to deliver reasonable results, but it was not possible to test this quantitatively as a method to directly compare a forecast to actuals was not found. This was also quite computationally intensive.

Expanding on the work in this project to forecast UK rainfall, other ANN algorithms, such as backpropagation, could be explored as well as other modelling and forecasting techniques including Support Vector Machines, for example.

References

[1] Wiener, N. (1949). Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications. Technology Press of the Massachusetts Institute of Technology

[2] Box, G.E.P. & Jenkins, G. M. (1970). Time series analysis : forecasting and control. Holden-Day

[3] Mitrea, C. A., Lee, C. K. M., Wu, Z. (2009). A Comparison between Neural Networks and Traditional Forecasting Methods: A Case Study. International Journal of Engineering Business Management, Vol. 1, No. 2, p 19-24

[4] Benkachacha, S., Benhra, J. & El Hassani, H. (2015). Seasonal Time Series Forecasting Models based on Artificial Neural Network. International Journal of Computer Applications, Vol. 116, No. 20

[5] Zhang, P.G. & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. European Journal of Operational Research, 160, p 501-514

[6] Met Office (2022). UK and regional series. Available at https://www.metoffice.gov.uk/research/climate/maps-and-data/uk-and-regional-series. Accessed 04/04/2022

[7] Hawkins, E., Burt, S., McCarthy, M., Murphy, C., Ross, C., Baldock, M., et al (2022) Millions of historical monthly rainfall observations taken in the UK and Ireland rescued by citizen scientists. Geoscience Data Journal, 00, 1– 16. Available from: https://doi.org/10.1002/gdj3.157

[8] Met Office (2022). UK climate districts map. Available at https://www.metoffice.gov.uk/research/climate/maps-and-data/about/districts-map. Accessed 04/04/2022

[9] Open Data NI (2022). OSNI Open Data – 50K Boundaries – NI Outline. Available at https://www.opendatani.gov.uk/dataset/osni-open-data-50k-boundaries-ni-outline. Accessed 04/04/2022

[10] Lee, E.M. (2020). Statistical analysis of long-term trends in UK effective rainfall: implications for deep-seated landsliding. Quarterly Journal of Engineering Geology and Hydrogeology, 53, 587-597

[11] Jenkins, G.J., Perry, M.C. & Prior, M.J. (2009). The Climate of the United Kingdom and Recent Trends. Met Office Hadley Centre, Exeter

[12] Hyndman, R.J., & Athanasopoulos, G. (2018). Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

Code Transcript and instructions for downloading R Project and associated data files

The R Project for this assignment, which includes this R script and all relevant data files, can be found in the following GitHub repository:

https://github.com/rubble35/STDM-coursework

```
# Install packages and libraries
install.packages("maptools")
install.packages("forecast")


library(sp)
library(maptools)
library(lattice)
library(spdep)
library(rgdal)
library(tmap)
library(ggplot2)
library(gridExtra)
library(scatterplot3d)
library(plot3D)
library(reshape)
library(rgl)
library(gstat)
library(OpenStreetMap)
library(raster)
library(spacetime)
library(forecast)
library(nnet)



## 1 IMPORT THE DATA

# 1.1 Monthly UK Rainfall Data by Region

# Dataset being used is monthly rainfall records for the UK sourced from the UK Met Office.
# This data was recently updated as a result of the crowdsourcing project 'Rainfall Rescue'.
# The period covered is January 1836 to December 2021.

## THIS IS A LARGE DATASET THAT TAKES SOME TIME TO LOAD ##

# Read geodatabase and set CRS using a proj4string
uk_rain_raw <- readOGR(dsn="Data/UK Rainfall.gdb", layer="uk_rain_all_districts",
                   p4s = CRS("+proj=tmerc +lat_0=49 +lon_0=-2 +k=0.9996012717 +x_0=400000 +y_0=-100000 +ellps=airy +datum=OSGB36 +units=m +no_defs")@projargs)

# Dataframe has 10 elements/features: 1 for each district region of the UK as defined by the MetOffice.
# Dataframe has 2232 columns/fields of data: one for each month for 186 years.

summary(uk_rain_raw)

# Remove the Shape_Area and Shape_Length fields from dataframe to avoid skewing rainfall data with these large values
uk_rain <- subset(uk_rain_raw, select = -c(Shape_Length, Shape_Area))
```

```
# Create a matrix and add row names
rain_matrix <- data.matrix(uk_rain@data[,-c(1)])
rownames(rain_matrix) <- uk_rain@data[,"REGION"]

# Visualise the data for one region by way of example
# Set parameters so that axis labels show
par(mar = c(5, 5, 4, 2) + 0.1)
plot(rain_matrix[1,], type="l", xaxt="n", xlab="Year (monthly data)", ylab="Monthly rainfall (mm)", main="Monthly Rainfall for Scotland N region, 1836 to 2021")
axis(1, at=seq(49, 2209, 240), labels=seq(1840, 2020, 20))
# Very long time series, so hard to see detail of any patterns or trends.
# Look at a shorter time period
plot(rain_matrix[1, 1969:2232], type="l", xaxt="n", xlab="Year (monthly data)", ylab="Monthly rainfall (mm)", main="Monthly Rainfall for Scotland N region, Jan 2000 to Dec 2021")
axis(1, at=seq(1, 264, 24), labels=seq(2000, 2020, 2))
# Would expect a seasonal trend, and it looks like there is one, but it is noisy
# Have a look at another region
plot(rain_matrix[6, 1969:2232], type="l", xaxt="n", xlab="Year (monthly data)", ylab="Monthly rainfall (mm)", main="Monthly Rainfall for East Anglia region, Jan 2000 to Dec 2021")
axis(1, at=seq(1, 264, 24), labels=seq(2000, 2020, 2))
# Also looks quite noisy, with significant variation between max and min and from year to year

# 1.2 Monthly UK Rainfall Records for Historic UK Weather Stations

# This dataset comprises historic monthly rainfall records for 37 individual weather stations across the UK sourced from the UK Met Office.
# The stations have latitude, longitude and altitude data, i.e. they are point locations rather than areas
# Data goes back to 1855 for some stations, but most stations have full records going back only to 1960's.
# Data imported for analysis is monthly for the period Sep 1964 to Aug 2016 (624 months), but just for 29 stations that have a full record in this timeframe.
uk_station <- read.csv("Data/rainfall_by_station.csv")

# Rename first column to remove strange characters at front
colnames(uk_station)[1] <- "Station"

# Create a matrix excluding the station name, lat, long and altitude columns
station_matrix <- data.matrix(uk_station[,-c(1:4)])

# Create a vector containing the station names and add to matrix
station <- as.vector(uk_station[,c(1)], mode="any")
rownames(station_matrix) <- station

# Visualise the data for one weather station by way of example
plot(station_matrix[1,], type="l", xaxt="n", xlab="Year (monthly data)", ylab="Rainfall (mm)", main="Average Monthly Rainfall for Aberporth, 1965 to 2015")
axis(1, at=seq(5, 605, 60), labels=seq(1965, 2015, 5))

# Also bring in file containing annual average rainfall for the same 29 weather stations for period 1965 to 2015
# And rename columns and create a data matrix
uk_station_annual <- read.csv("Data/station_annual_ave.csv")
colnames(uk_station_annual)[1] <- "Station"
colnames(uk_station_annual)[5:ncol(uk_station_annual)] <- as.character(c(1965:2015))
station_annual_matrix <- data.matrix(uk_station_annual[,-c(1:4)])
rownames(station_annual_matrix) <- station

# Visualise the data for one weather station by way of example
```

```
plot(station_annual_matrix[1,], type="l", xaxt="n", xlab="Year", ylab="Rainfall (mm)", main="Average Annual Rainfall for Aberporth, 1965 to 2015")
axis(1, at=seq(1, 51, 5), labels=seq(1965, 2015, 5))
# Very variable from one month to the next. Possibly seasonal as there are a series of peaks and troughs

# Create column averages and plot to see any trends in average annual rainfall for whole of UK
plot(colMeans(station_annual_matrix), type="l", xaxt="n", xlab="Year", ylab="Rainfall (mm)", main="Average Annual Rainfall for all UK Weather Stations, 1965 to 2015")
axis(1, at=seq(1, 51, 5), labels=seq(1965, 2015, 5))
# No obvious long-term trend in average annual rainfall


# 1.3 Examine non-spatio-temporal data characteristics

# Mean and standard deviation for regional data
mean_rain <- mean(rain_matrix)
mean_scotland_n <- mean(rain_matrix[1,])
mean_scotland_e <- mean(rain_matrix[2,])
mean_scotland_w <- mean(rain_matrix[3,])
mean_england_e_ne <- mean(rain_matrix[4,])
mean_england_nw_n_wales <- mean(rain_matrix[5,])
mean_midlands <- mean(rain_matrix[6,])
mean_e_anglia <- mean(rain_matrix[7,])
mean_s_wales_england_sw <- mean(rain_matrix[8,])
mean_england_se_central_s <- mean(rain_matrix[9,])
mean_n_ireland <- mean(rain_matrix[10,])
sd_rain <- sd(rain_matrix)
mean_rain   # Mean monthly rainfall for all UK regions across all years
sd_rain

# Mean and standard deviation for weather station data
mean_station <- mean(station_matrix)
sd_station <- sd(station_matrix)
mean_station    # Mean monthly rainfall recorded at UK selected weather stations across all years
sd_station

# Histograms
hist(rain_matrix, col="lightblue", xlab="Monthly Rainfall in mm", main="Histogram of UK Regional Monthly Rainfall")
abline(v=mean_rain, col="red", lwd=3)
text(120, 3849, "Mean = 88.55")
# Shows positive skew, bounded by zero (cannot have -ve rainfall)

hist(station_matrix, col="lightgreen", xlab="Monthly Rainfall in mm", main="Histogram of UK Weather Station Monthly Rainfall")
abline(v=mean_station, col="red", lwd=3)
text(120, 7650, "Mean = 70.20")
# Shows positive skew, bounded by zero (cannot have -ve rainfall)

# Q-Q Plots
qqnorm(rain_matrix)
qqline(rain_matrix, col="red", lwd=3)
qqnorm(station_matrix)
qqline(station_matrix, col="red", lwd=3)


## 2 EXPLORATORY SPATIO-TEMPORAL DATA ANALYSIS AND VISUALISATION
```

```
# 2.1. Examining Temporal Characteristics

# First examine the average monthly rainfall, both on the regional dataset and the individual station dataset
# Regional dataset, monthly granularity, 186 years, average rainfall across all regions
plot(colMeans(rain_matrix), xlab="Year (monthly data)", ylab="Rainfall in mm", type="l", xaxt="n", main="Average
UK Monthly Rainfall 1836-2021")
axis(1, at=seq(49, 2209, 240), labels=seq(1840, 2020, 20))
# No obvious trends - data too dense to really see - significant variation across the months within a year
# Suggest using annual averages?

# Station dataset, monthly granularity, 52 years, average rainfall across all weather stations
plot(colMeans(station_matrix), xlab="Year (monthly data)", ylab="Rainfall (mm)", type="l", xaxt="n", main="Average
Monthly Rainfall for Selected UK Weather Stations 1965-2015")
axis(1, at=seq(5, 605, 120), labels=seq(1965, 2015, 10))
# No obvious trends - data too dense to really see - significant variation across the months within a year
# Hint of a seasonal pattern

# Try looking at a shorter time range: 2005-2015
plot(colMeans(station_matrix[,504:624]), xlab="Year (monthly data)", ylab="Rainfall (mm)", type="l", xaxt="n",
main="Average Monthly Rainfall for Selected UK Weather Stations 2005-2015")
axis(1, at=seq(1, 121, 12), labels=seq(2005, 2015, 1))
# No obvious long-term trend - significant variation across the months within a year
# Looks like a potentially seasonal pattern

# Suggest using annual averages:
plot(colMeans(station_annual_matrix), xlab="Year", ylab="Rainfall (mm)", type="l", xaxt="n", main="Average Annual
Rainfall for Selected UK Weather Stations 1965-2015")
axis(1, at=seq(1,51,5), labels=seq(1965, 2015, 5))
# Possible trend of increased rainfall with time, but extent of variation from year to year dominates

# Create Lattice Plot for 10 selected stations that cover all parts of the UK
# Rainfall to be dependent variable
# Month of the year (MMM.YY) the independent variable
station_melt <- melt(uk_station, id.vars=1:4, measure.vars = 5:ncol(uk_station))
colnames(station_melt)[5:6] <- c("MMMYY", "Rainfall")
station.chosen=c("Aberporth","Armagh","Bradford","Braemar","Cambridge","Oxford","Yeovilton","Lowestoft","Lerw
ick","Durham")
s <- station_melt[station %in% station.chosen,]
xyplot(Rainfall ~ MMMYY | Station, xlab = "MMM.YY", type ="l",
    layout = c(5,2),
    data = s,
    main = "Monthly Rainfall at Selected UK Weather Stations Sep 1965 to Aug 2016")
# No obvious trends here either, although possible increase over time seen in Lerwick?
# Data possibly too granular

# Use annual averages instead
station_annual_melt <- melt(uk_station_annual, id.vars=1:4, measure.vars = 5:ncol(uk_station_annual))
colnames(station_annual_melt)[5:6] <- c("Year", "Rainfall")
sam <- station_annual_melt[station %in% station.chosen,]
xyplot(Rainfall ~ Year | Station, xlab = "Year", type ="l",
    layout = c(5,2),
    data = sam,
    main = "Annual Average Rainfall at Selected UK Weather Stations 1965-2015")
# Data easier to interpret than monthly data, but still no consistent temporal trends
# Some hint at slightly increased rainfall over time, e.g.Lerwick and Braemar
```

```r
# 2.3 Examining Spatial Characteristics

# Scatterplot Matrix for UK Weather Stations
pairs(~LONG+LAT+ALT+rowMeans(station_matrix),data=uk_station,main="Simple Scatterplot Matrix for Rainfall at UK Weather Stations")
# bottom left plot hints at a relationship between longitude and rainfall - further west, higher rainfall
# furthest right plot on second row also hints at a relationship between latitude and rainfall - further south, lower rainfall
# but trends are not clear

# 3D Scatterplots
scatterplot3d(x=uk_station$LONG, y=uk_station$LAT, z=rowMeans(station_matrix), main="3D Scatterplot of Rainfall at UK Weather Stations by Longitude and Latitude", xlab="Longitude", ylab="Latitude", zlab="Average Monthly Rainfall in mm")
scatter3D(uk_station$LONG, uk_station$LAT, rowMeans(station_matrix), main="3D Scatter of Rainfall at UK Weather Stations", xlab="Longitude", ylab="Latitude", zlab="Average Monthly Rainfall in mm")
plot3d(uk_station$LONG, uk_station$LAT, rowMeans(station_matrix), main="Dynamic 3D Plot of Rainfall at UK Weather Stations", xlab="Longitude", ylab="Latitude", zlab="Average Monthly Rainfall in mm")
# 3D plot confirms that there appears to be a relationship between latitude, longitude and rainfall, in that the further north and west, the higher the average monthly rainfall. But pattern is not clear

# Heatmap
heatmap(station_matrix,Rowv=NA,Colv=NA, col=cm.colors(256), scale="column", margins=c(5,3), xlab="MMM.YY", ylab="Station", main="Heatmap of Rainfall at UK Weather Stations, Unordered", cexCol=1.1, y.scale.components.subticks(n=10))
# Shows that there is not much temporal variation, as each row shows limited colour variation
# Try ordering stations by latitude to see if that shows anything
station_latorder <- uk_station[order(uk_station$LAT, decreasing=FALSE),]
station_latorder_matrix <- data.matrix(station_latorder[,5:ncol(uk_station)])
heatmap(station_latorder_matrix,Rowv=NA,Colv=NA, col=cm.colors(256), scale="column", margins=c(5,3), xlab="MMM.YY", ylab="Station (N at the top, S at the bottom)", main="Heatmap of Rainfall at UK Weather Stations, Ordered by Latitude", cexCol=1.1, y.scale.components.subticks(n=10))
# Definitely shows pinker colours (most rainfall) at the top (highest latitudes), i.e. furthest north
# Try ordering by longitude as well
station_longorder <- uk_station[order(uk_station$LONG, decreasing=TRUE),]
station_longorder_matrix <- data.matrix(station_longorder[,5:ncol(uk_station)])
heatmap(station_longorder_matrix,Rowv=NA,Colv=NA, col=cm.colors(256), scale="column", margins=c(5,3), xlab="MMM.YY", ylab="Station (W at the top, E at the bottom)", main="Heatmap of Rainfall at UK Weather Stations, Ordered by Longitude",cexCol=1.1, y.scale.components.subticks(n=10))
# Less obvious than latitude heat map, but suggests pinker colours (most rainfall) at the top (lowest longitudes), i.e. furthest west
# Station 12 is a bit of an anomaly (this is Lerwick, the furthest north by far in the Shetlands)
# Suggests latitude is more of a factor that longitude, but this might be expected as UK has a greater N/S extent than E/W

# Plot annual average rainfall for weather stations on a map
ann_ave <- cbind(uk_station_annual[1:4], rowMeans(station_annual_matrix))
colnames(ann_ave)[5] <- "Ave.Rainfall"
ann_ave[,2:3] <- projectMercator(ann_ave$LAT, ann_ave$LONG)
map <- openmap(c(49,-11), c(61,3), type='esri-topo')
autoplot.OpenStreetMap(map) +
  geom_point(data = ann_ave, aes(x = LONG, y = LAT, color = Ave.Rainfall, size = Ave.Rainfall)) +
  ggtitle("Annual Average Rainfall in UK, all years") +
  scale_color_gradient(low="lightblue", high="darkblue")
# Shows wettest areas are in the north and west; drier to south and east and away from coasts
# Eskdalemuir records most rain annually out of the 29 weather stations in the dataset
# Going back to the regional data
# Create breaks that can be used consistently in each map based on entire data set
brks=quantile(as.numeric(unlist(uk_rain@data[,-c(1)])), seq(0,1,1/5))
# Produce a choropleth map of UK rainfall by region for March 1990, which based on analysis in Excel is the month that shows the greatest contrast between wettest and driest region
tm_shape(uk_rain) +
  tm_fill("Mar_1990", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_compass(position=c("left", "bottom")) +
  tm_legend(position=c("left", "top")) +
  tm_scale_bar()
# Very wet in Scotland, particularly in the N and W; dry elsewhere

# Produce a choropleth map of UK rainfall by region for Nov 1907, which based on analysis in Excel is the month that is closest to the overall average rainfall for each region
tm_shape(uk_rain) +
  tm_fill("Nov_1907", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_compass(position=c("left", "bottom")) +
  tm_legend(position=c("left", "top")) +
  tm_scale_bar()
# Wettest in the West, driest in the East

# Whilst temperature follows fairly distinct seasonal patterns (warmer in summer, colder in winter), rainfall in the UK may not
# Produce choropleth maps for each month in 1984, which is identified as the year that is closest to the overall annual average rainfall for each region to see if there are any hints of seasonal patterns
year84 <- subset(uk_rain, select = c(Jan_1984:Dec_1984))
jan84 <- tm_shape(year84) +
  tm_fill("Jan_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_compass(position=c("right", "top"), size=1) +
  tm_legend(position=c("left", "top"), text.size=0.4) +
  tm_scale_bar()

feb84 <- tm_shape(year84) +
  tm_fill("Feb_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

mar84 <- tm_shape(year84) +
  tm_fill("Mar_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

apr84 <- tm_shape(year84) +
  tm_fill("Apr_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

may84 <- tm_shape(year84) +
  tm_fill("May_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

jun84 <- tm_shape(year84) +
  tm_fill("Jun_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

jul84 <- tm_shape(year84) +
  tm_fill("Jul_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

aug84 <- tm_shape(year84) +
  tm_fill("Aug_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

sep84 <- tm_shape(year84) +
  tm_fill("Sep_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

oct84 <- tm_shape(year84) +
  tm_fill("Oct_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

nov84 <- tm_shape(year84) +
  tm_fill("Nov_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

dec84 <- tm_shape(year84) +
  tm_fill("Dec_1984", style="fixed", palette="Blues", breaks=brks) +
  tm_borders("white") +
  tm_legend(position=c("left", "top"), text.size=0.4)

tmap_arrange(jan84, feb84, mar84, apr84, may84, jun84, jul84, aug84, sep84, oct84, nov84, dec84) # This takes a long time to plot

# These maps clearly show that (for this single year) the western and northern regions of the UK are the wettest, although N.Ireland is relatively drier than nearby W Scotland
# The driest regions are towards the east and the south
# This is in line with expectations based on knowledge of the south-westerly prevailing weather pattern in the UK, which brings rain in from the Atlantic and the central hills and mountains of England and Scotland create a rain shadow effect to the east
# Looking just at 1984, there is an indication of seasonality with more rain in the autumn/winter than spring/summer.
# Needs investigating further to see if this is a pattern that bears out across more/all years in the study.


# 2.3 Spatial and Temporal Autocorrelation

# 2.3.1. Spatial Autocorrelation

# Build a row standardised spatial weight matrix for the UK Regions (using Queen's case)
W <- nb2listw(poly2nb(uk_rain, queen=TRUE))
# Error due to empty neighbour sets found
# Test whether this is due to N.Ireland, which is not connected to rest of UK
W <- nb2listw(poly2nb(uk_rain[1:9,], queen=TRUE))
```

```
# Success - it was due to N.Ireland
W

# 2.3.1.1 Global Spatial Autocorrelation

# Calculate Moran's I

# This is a purely spatial index so can just take average rainfall for whole period under review
# Need to create subset of uk_rain that excludes N.Ireland due to empty neighbour sets issue above
gb_rain <- uk_rain[1:9,] # excluding N.Ireland
gb_rain_matrix <- data.matrix(gb_rain@data[,-c(1)])
gb_rain_ave <- rowMeans(gb_rain_matrix) # excluding N.Ireland
moran(gb_rain_ave, W, n=length(W$neighbours), S0=Szero(W))
# Gives a Moran's I value of 0.516

# Test min and max values to see whether reasonable on an absolute scale
moran.range <- function(lw) {
  wmat <- listw2mat(lw)
  return(range(eigen((wmat + t(wmat))/2)$values))
}
moran.range(W)
# Moran range is approx -0.6 to 1.0, with mid point of approx 0.2.
# Therefore, Moran's I of 0.516 appears significant

# Test whether it is statistically significant using moran.test and moran.mc
moran.test(x=gb_rain_ave, listw=W)
moran.mc(x=gb_rain_ave, listw=W, nsim=9999)
# Both tests give p-values <0.05, so can conclude that there is significant spatial autocorrelation between rainfall levels across the regions of the UK at this spatial order

# 2.3.1.2 Local Spatial Autocorrelation

# Calculate Local Moran's I for Regional data
local_m <- localmoran(x=gb_rain_ave, listw=W)
gb_rain$Ii <- local_m[,"Ii"]
gb_rain$Iip_unadj <- local_m[,"Pr(z != E(Ii))"]
gb_rain$Unadjusted_significance <- "nonsignificant"
gb_rain$Unadjusted_significance[which(gb_rain$Iip_unadj < 0.05)] <- "significant"
tm_shape(gb_rain) +
  tm_polygons(col="Unadjusted_significance", palette="-RdBu", style="quantile", lwd=0.1, border.alpha=0.1) +
  tm_layout(main.title="Local Moran's I by Region", main.title.size=1.25, title="P-values unadjusted", title.size=1,legend.position=c("left", "top")) +
  tm_scale_bar(width=0.25, position=c("right", "bottom")) +
  tm_compass(position=c("right", "top"))
# "S Wales & England SW" and "England NW & N Wales" are the regions with the lowest local Moran values (close to zero)
# These regions are wet regions, being in the West, but they border the driest regions, i.e. those in the south and east
# "East Anglia" is a dry region bordering other dry regions, hence high local Moran
# "Scotland N" is a wet region bordering other wet regions, hence high local Moran
# "East Anglia" and "Scotland W" have significant local Moran's I based on unadjusted p-values

local_m_adj <- localmoran(x=gb_rain_ave, listw=W, p.adjust.method="bonferroni")
gb_rain$Iip_adj <- local_m_adj[,"Pr(z != E(Ii))"]
gb_rain$Adjusted_significance <- "nonsignificant"
gb_rain$Adjusted_significance[which(gb_rain$Iip_adj <0.05)] <- "significant"
tm_shape(gb_rain) +
  tm_polygons(col="Adjusted_significance", palette="-RdBu", style="quantile", lwd=0.1, border.alpha=0.1) +
  tm_layout(main.title="Local Moran's I by Region", main.title.size=1.25, title="P-values adjusted", title.size=1, legend.position=c("left", "top")) +
  tm_scale_bar(width=0.25, position=c("right", "bottom")) +
  tm_compass(position=c("right", "top"))
# No regions with significant local Moran's I based on adjusted p-values

# Measure autocorrelation in weather station point data using semivariogram
# Use average rainfall across all time periods under review as temporal variation not taken into account in semivariogram
coords = list(projectMercator(uk_station[,3], uk_station[,2]))
plot(variogram(list(rowMeans(station_matrix)), location=coords), cex=2, main="Semivariogram for UK Weather Station Rainfall", col="blue")
# Result is a bit scattered, but can conclude that rainfall levels at weather stations that are closer are more similar than those further away

# Consider different directions
plot(variogram(list(rowMeans(station_matrix)), location=coords, alpha=c(0,45,90,135)), cex=1.5, main="Directional semivariograms for UK Weather Station Rainfall", col="blue")
# Not very clear results, but hints of anisotropy in that not all semivariograms are the same


# 2.3.2 Temporal Autocorrelation

# Using the monthly rainfall data by region
# Create dataframes containing lagged variables at lag of 1 month
# Just describing the time intervals as 'time_in_months' and giving them a numerical index as cannot find a way to convert the format in the original data to a date that the formula will accept
Scotland_N_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[1,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[1,][1:(ncol(rain_matrix)-1)])
Scotland_E_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[2,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[2,][1:(ncol(rain_matrix)-1)])
Scotland_W_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[3,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[3,][1:(ncol(rain_matrix)-1)])
England_E_NE_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[4,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[4,][1:(ncol(rain_matrix)-1)])
England_NW_N_Wales_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[5,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[5,][1:(ncol(rain_matrix)-1)])
Midlands_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[6,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[6,][1:(ncol(rain_matrix)-1)])
East_Anglia_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[7,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[7,][1:(ncol(rain_matrix)-1)])
S_Wales_England_SW_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[8,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[8,][1:(ncol(rain_matrix)-1)])
England_SE_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[9,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[9,][1:(ncol(rain_matrix)-1)])
N_Ireland_lagged <- data.frame(time_in_months = 1:2231, t=rain_matrix[10,][2:(ncol(rain_matrix))], t_minus_1=rain_matrix[10,][1:(ncol(rain_matrix)-1)])

# Calculate PMCC based on 1 month lags
SN_r <- round(cor(Scotland_N_lagged$t, Scotland_N_lagged$t_minus_1), 3) # gives 0.306
SE_r <- round(cor(Scotland_E_lagged$t, Scotland_E_lagged$t_minus_1), 3) # gives 0.152
SW_r <- round(cor(Scotland_W_lagged$t, Scotland_W_lagged$t_minus_1), 3) # gives 0.280
EENE_r <- round(cor(England_E_NE_lagged$t, England_E_NE_lagged$t_minus_1), 3) # gives 0.091
EWNW_r <- round(cor(England_NW_N_Wales_lagged$t, England_NW_N_Wales_lagged$t_minus_1), 3) # gives 0.180
M_r <- round(cor(Midlands_lagged$t, Midlands_lagged$t_minus_1), 3)  # gives 0.081
```

```
EA_r <- round(cor(East_Anglia_lagged$t, East_Anglia_lagged$t_minus_1), 3) # gives 0.103
EWSW_r <- round(cor(S_Wales_England_SW_lagged$t, S_Wales_England_SW_lagged$t_minus_1), 3) # gives 0.178
ESE_r <- round(cor(England_SE_lagged$t, England_SE_lagged$t_minus_1), 3)  # gives 0.127
NI_r <- round(cor(N_Ireland_lagged$t, N_Ireland_lagged$t_minus_1), 3) # gives 0.135

# Most of these PMCCs show only weak or very weak correlation at 1 month lag interval
# Scotland N shows the highest correlation at 0.306
# Plot to visualise
p1 <- ggplot(Scotland_N_lagged, aes(x=time_in_months, y=t)) + geom_line()
p2 <- ggplot(Scotland_N_lagged, aes(x=t, y=t_minus_1)) +
  geom_point() +
  labs(y="t-1") +
  geom_smooth(method="lm") +
  annotate("text", 30, 320, label=paste("r =", SN_r))

grid.arrange(p1,p2, nrow=1)

# Now just plot Scotland N PMCC
ggplot(Scotland_N_lagged, aes(x=t, y=t_minus_1)) +
  geom_point() +
  labs(y="t-1") +
  geom_smooth(method="lm") +
  annotate("text", 30, 320, label=paste("r =", SN_r))

# Also try looking at 12month lag
Scotland_N_lagged12 <- data.frame(time_in_months = 12:2231, t=rain_matrix[1,][13:(ncol(rain_matrix))],
t_minus_12=rain_matrix[1,][1:(ncol(rain_matrix)-12)])
SN_r_minus12 <- round(cor(Scotland_N_lagged12$t, Scotland_N_lagged12$t_minus_12), 3)
ggplot(Scotland_N_lagged12, aes(x=t, y=t_minus_12)) +
  geom_point() +
  labs(y="t-12") +
  geom_smooth(method="lm") +
  annotate("text", 30, 320, label=paste("r =", SN_r_minus12))
# PMCC of 0.331 implies stronger temporal correlation at 12 month lag than 1 month lag
# Suggests rainfall in a given month is more similar to the same month in the previous year than in the previous
month.
# Suggests rainfall varies more month-to-month than year-to-year in Scotland N region

# Next, examining annual averages by weather station
# First create a dataset of annual averages across all weather stations
uk_station_annual_aves <- colMeans(station_annual_matrix)
uk_annual_lagged <- data.frame(year = 1965:2014, t=uk_station_annual_aves[2:(length(uk_station_annual_aves))],
t_minus_1=uk_station_annual_aves[1:(length(uk_station_annual_aves)-1)])
uk_r <- round(cor(uk_annual_lagged$t, uk_annual_lagged$t_minus_1), 3)

# Plot to visualise
p3 <- ggplot(uk_annual_lagged, aes(x=year, y=t)) + geom_line()
p4 <- ggplot(uk_annual_lagged, aes(x=t, y=t_minus_1)) +
  geom_point() +
  labs(y="t-1") +
  geom_smooth(method="lm") +
  annotate("text", 60, 90, label=paste("r =", uk_r))

grid.arrange(p3,p4, nrow=1)
# Shows that annual average rainfall is weakly correlated at a lag of one year (PMCC=0.121)
# This is not surprising as no reason to suggest that rainfall in one year is related to rainfall in previous year
# 2.3.3 Spatio-Temporal Autocorrelation

# Import starima_package
source("starima_package.R")

# Attempt space-time semivariogram using UK weather station point data
# Project points to get spatial lag in metres
points <- SpatialPoints(uk_station_annual[,2:3], proj4string = CRS("+init=epsg:4326 +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0"))
# Convert years to correct date format
years <- seq(as.Date("1965-01-01"), length=51, by="year")
# Create spatio-temporal data frame
stfdf <- STFDF(points, years, data.frame(as.vector(t(station_annual_matrix))))
names(stfdf@data) <- "Rainfall"

# Calculate space-time semivariogram and plot
station_ST_var <- variogram(Rainfall~1, stfdf, width=50, cutoff=500, tlags=0:10)
plot(station_ST_var)
plot(station_ST_var, wireframe=T)
# What does this show?
# Semivariogram increases much more rapidly with increasing temporal lag than with increasing spatial lag
# But time and space are not the same thing so cannot directly compare these 'distances'


## 3 STATISTICAL MODELLING OF TIME SERIES AND SPATIO-TEMPORAL SERIES

# 3.1 Time Series Decomposition

# Use STL to decompose the Scotland N data
# Choosing this dataset as it showed the highest levels of temporal autocorrelation
# First, separate the Scotland N data from the rest of the UK rain data and convert to a time-series
Scot_N_ts <- ts(rain_matrix[1,], frequency = 12, start=c(1836,1))

# Then run stl function (t.window parameter should be odd or NULL)
decom <- stl(Scot_N_ts, t.window=NULL, s.window="periodic")
autoplot(decom)
# Hard to see due to length of timeseries

# Try on shorter timeseries (1990-2021)
Scot_N_sts <- ts(rain_matrix[1,1849:2232], frequency = 12, start=c(1990,1))
decom <- stl(Scot_N_sts, t.window=25, s.window="periodic")
autoplot(decom)
# Various different values of t.window used to extract a trend. Settled for t.window=25 as remainders smaller
# Trend component not clear - may not need nonseasonal differencing; may only need to use seasonal differencing
# Clear seasonality extracted through the stl decomposition process, although seasonal component only ranges +/-60mm
# Remainders still significant in size, typically ranging +/-100mm - larger than the seasonal component (which is actually smaller than the trend component based on the bars to the side of the plots)


# 3.2 ARIMA

# Use the Box-Jenkins approach to ARIMA modelling

# 3.2.1 Exploratory Data Analysis
```

```r
# Have already performed this above
# UK Regional rainfall data shows seasonality as it is presented as monthly, with no clear trend up or down over time
# Annual average rainfall for UK Weather Stations does not show seasonality as it is annual, but no clear trend up or down over time: it appears stationary

# Lag plot for Scotland N region
lag.plot(rain_matrix[1,], lags=12, do.lines=FALSE)
# No clear patterns identified, which is a bit surprising considering seasonality already identified

# 3.2.2 Differencing, AutoCorrelation and Partial Autocorrelation Factors

# Examine the Scotland N time series and associated ACF and PACF
# ggtsdisplay function sourced from Hyndman and Athanasopoulos (2018) Forecasting: Principles and Practice, 2nd Ed. otexts.com/fpp2/
ggtsdisplay(Scot_N_sts, main="Scotland N timeseries 1990 to 2021")
# ACF shows strong seasonal pattern with positive peaks at lags 0 (as per default), 12, 24, 36, 48 and negative peaks at lags 6, 18, 30, 42
# Suggests seasonal differencing is required with an order of 12, which makes sense as this is monthly data

# Seasonal differencing
# Perform first seasonal differencing at lag 12, plot the differenced data and re-run the ACF and PACF
SN_s_diff <- diff(Scot_N_sts, lag=12)
ggtsdisplay(SN_s_diff, main="Scotland N Seasonally Differenced")
# Seasonal differencing has pretty much removed the seasonal pattern after lag 12
# Fits description of "one or more spikes, rest essentially zero", which implies MA model
# Significant autocorrelation remains at lags 1 and 12 (seasonal lag 1), which suggests Q=1 and potentially q=1
# Significant PACF values seen at lag 1 and seasonal lags 1, 2, 3 - potentially suggests p=1?

# Try non-seasonal differencing and re-run ACF and PACF
SN_ns_diff <- diff(Scot_N_sts)
ggtsdisplay(SN_ns_diff, main="Scotland N Non-Seasonally Differenced")
# ACF shows significant -ve ACF at lag 1, 6, 18 and +ve ACF at lags 12, 24, which suggests seasonal pattern remains
# This suggests non-seasonal differencing alone does not lead to stationarity
# PACF shows significant -ve PACF at lag 1, which slowly decays to insignificant values, indicative of an MA model (q=1)

# Try seasonal and nonseasonal differencing together
SN_sns_diff <- diff(diff(Scot_N_sts, lag=12))
ggtsdisplay(SN_sns_diff, main="Scotland N Non-Seasonally and Seasonally Differenced")
# ACF shows spikes at lag 1, 11, 13 and seasonal lag 1
# PACF shows exponential decay of -ve values for non-seasonal lags and +ve values for seasonal lags-1

# Seasonal differencing on its own seems to provide the clearest suggestions for a seasonal ARIMA model


# Examine the ACF and PACF plots together
p5 <- autoplot(acf(SN_s_diff, lag.max=36, plot=FALSE)) + ggtitle("ACF, Scotland N Seasonally Differenced")
p6 <- autoplot(pacf(SN_s_diff, lag.max=36, plot=FALSE)) + ggtitle("PACF, Scotland N Seasonally Differenced")
p7 <- autoplot(acf(SN_ns_diff, lag.max=36, plot=FALSE)) + ggtitle("ACF, Scotland N NonSeasonally Differenced")
p8 <- autoplot(pacf(SN_ns_diff, lag.max=36, plot=FALSE)) + ggtitle("PACF, Scotland N NonSeasonally Differenced")
p9 <- autoplot(acf(SN_sns_diff, lag.max=36, plot=FALSE)) + ggtitle("ACF, Scotland N Seasonally and NonSeasonally Differenced")
p10 <- autoplot(pacf(SN_sns_diff, lag.max=36, plot=FALSE)) + ggtitle("PACF, Scotland N Seasonally and NonSeasonally Differenced")
grid.arrange(p5,p6,p7,p8,p9,p10)
```

```
# 3.2.2.1 Spatio-Temporal ACF and PACF

# Calculate spatio-temporal autocorrelation factors (stacf) using UK regional (areal) data (excluding N.Ireland)
weight_matrix <- listw2mat(W)
stacf(t(gb_rain_matrix), weight_matrix, 48)
# Shows seasonal pattern with peaks at lag 1, 13, 25, 37
# and troughs at lag 7, 19, 30, 43

# Calculate spatio-temporal partial autocorrelation factors (stpacf) using UK regional (areal) data (excluding
N.Ireland)
stpacf(t(gb_rain_matrix), weight_matrix, 12)
# stpacf is insignificant for all temporal lags - implies the data are essentially random




# 3.2.3. Parameter Estimation and Fitting

# Analysis of ACF and PACF following seasonal differencing suggests a model of ARIMA(0,0,1)(0,1,1)12 for Scotland N
region
# q=1 for non-seasonal MA due to significant lag at lag 1 in seasonally differenced ACF
# Q=1 for seasonal MA due to significant lag at seasonal lag 1 (lag 12) and nothing at e.g. lag=24 or 36
# D=1 as used 1 difference in seasonal differencing
# m=12 as monthly data

# Using Arima function instead of arima function based on recommendation in Hyndman and Athanasopoulos
fit.Ar <- Arima(rain_matrix[1,], order=c(0,0,1), seasonal=list(order=c(0,1,1), period=12))
fit.Ar
# Gives log likelihood of -11815.62, AICc of 23637.25

NRMSE_fit <- NRMSE(res=fit.Ar$residuals, obs=rain_matrix[1,])
NRMSE_fit
# gives figure of 0.84564

# 3.2.4 Diagnostic Checking

# Check the residuals to see if any autocorrelations remain
checkresiduals(fit.Ar)
# Autocorrelations are within threshold limits - No significant ACF, except at lag 18 - good - but perhaps could be
better
# Residuals appear normally distributed - good
# p-value of 0.5426 (>0.05) - good

tsdiag(fit.Ar)
# p values in Ljung-Box plot are all large, which suggests no significant lack of fit


# Before moving to prediction, check if there are any other models that yield better results
# As initial model but add a non-seasonal AR component
fit2.Ar <- Arima(rain_matrix[1,], order=c(1,0,1), seasonal=list(order=c(0,1,1), period=12))
fit2.Ar
checkresiduals(fit2.Ar)
NRMSE_fit2 <- NRMSE(res=fit2.Ar$residuals, obs=rain_matrix[1,])
# Results similar to initial model
# Significant ACF at lag 18
# As initial model but add a seasonal AR component
fit3.Ar <- Arima(rain_matrix[1,], order=c(0,0,1), seasonal=list(order=c(1,1,1), period=12))
fit3.Ar
checkresiduals(fit3.Ar)
NRMSE_fit3 <- NRMSE(res=fit3.Ar$residuals, obs=rain_matrix[1,])
# Based on AICc, this performs better than initial model
# Significant ACF at lag 18

# As last (best so far) model but add a second seasonal AR component
fit4.Ar <- Arima(rain_matrix[1,], order=c(0,0,1), seasonal=list(order=c(2,1,1), period=12))
fit4.Ar
checkresiduals(fit4.Ar)
NRMSE_fit4 <- NRMSE(res=fit4.Ar$residuals, obs=rain_matrix[1,])
# Better AICc value again, p-value deteriorating, but still >0.05
# Significant ACF at lag 18

# As last (best so far) model but add a non-seasonal AR component
fit5.Ar <- Arima(rain_matrix[1,], order=c(1,0,1), seasonal=list(order=c(2,1,1), period=12))
fit5.Ar
checkresiduals(fit5.Ar)
NRMSE_fit5 <- NRMSE(res=fit5.Ar$residuals, obs=rain_matrix[1,])
# Based on AICc, not as good as last model and p-value reducing further
# Significant ACF at lag 18
# Also get warning of NaNs for some of the coefficients

# As best so far model but add a non-seasonal AR component instead of non-seasonal MA component
fit6.Ar <- Arima(rain_matrix[1,], order=c(1,0,0), seasonal=list(order=c(2,1,1), period=12))
fit6.Ar
checkresiduals(fit6.Ar)
NRMSE_fit6 <- NRMSE(res=fit6.Ar$residuals, obs=rain_matrix[1,])
# Almost identical to fit4. AICc is 0.01 lower, which makes it very marginally better
# Significant ACF at lag 18

# As last model but remove seasonal AR components
fit7.Ar <- Arima(rain_matrix[1,], order=c(1,0,0), seasonal=list(order=c(0,1,1), period=12))
fit7.Ar
checkresiduals(fit7.Ar)
NRMSE_fit7 <- NRMSE(res=fit7.Ar$residuals, obs=rain_matrix[1,])
# Not as good as fit4 or fit6 or initial model
# Significant ACF at lag 18
# Almost identical diagnostic results as initial model, which suggests non-seasonal AR or non-seasonal MA makes little difference

# As last model but add back a seasonal AR component
fit8.Ar <- Arima(rain_matrix[1,], order=c(1,0,0), seasonal=list(order=c(1,1,1), period=12))
fit8.Ar
checkresiduals(fit8.Ar)
NRMSE_fit8 <- NRMSE(res=fit8.Ar$residuals, obs=rain_matrix[1,])
# Similar to fit4, but not as good as fit5 or fit7
# Significant ACF at lag 18



# 3.2.5 Prediction

# Train over first 176 years, test over last 10 years, using best fit model fit6.Ar
```

```
train.Ar <- Arima(rain_matrix[1,1:2112], order=c(1,0,0), seasonal=list(order=c(2,1,1), period=12))
pred.Ar <- Arima(rain_matrix[1,2113:ncol(rain_matrix)], model=train.Ar)
par(mar = c(5, 5, 4, 2) + 0.1)
matplot(cbind(pred.Ar$fitted, pred.Ar$x), type="l", xlab="months", ylab="Rainfall (mm)", main="ARIMA Prediction vs. Actuals for Scotland N, 2011 to 2021")
# Black line is prediction, red-dashed line is actuals
# Results pretty good for the first year or so, but then it breaks down somewhat
# Overall shape fairly accurate but looks like there is a trend to the amplitude of the seasonal peaks that increases in the first few years and then gradually decreases


# Have a look at auto.ar to see what it gives as a suggested ARIMA model and compare with my model
# Adapted from Hyndman and Athanasopoulos
fit.auto.ar <- auto.arima(rain_matrix[1,], seasonal=TRUE, stepwise=FALSE, approximation=FALSE)
fit.auto.ar
# Gives "ARIMA(3,1,2)"
# No seasonal component, which is strange as function settings specifically allow seasonal models
# Can't compare AICc due to different differencing component, but can check residuals
checkresiduals(fit.auto.ar)
tsdiag(fit.auto.ar)
NRMSE_fit.auto.ar <- NRMSE(res=fit.auto.ar$residuals, obs=rain_matrix[1,])
# This model fails the checks due to significant ACF spikes, which look seasonal, and p-value is <<0.05
# Ljung-Box plot from tsdiag shows p-values are pretty much all significant (i.e. <0.05)
# NRMSE is 0.92012, which is higher than any of the manual models tested
# Perhaps the seasonality was too weak to be picked up and first differencing achieved the necessary stationarity for the auto.Arima function to not need to look at seasonal differencing?
# This suggests that finding the right model is not straightforward


# Use auto.arima on training set and then use to predict
# Check what forecast versus actuals looks like for auto.arima results
train.auto.ar <- Arima(rain_matrix[1,1:2112], order=c(3,1,2))
pred.auto.ar <- Arima(rain_matrix[1,2113:ncol(rain_matrix)], model=train.auto.ar)
matplot(cbind(pred.auto.ar$fitted, pred.auto.ar$x), type="l", xlab="months", ylab="Rainfall (mm)", main="Auto.ARIMA Prediction vs. Actuals for Scotland N, 2011 to 2021")
# Black line is prediction, red-dashed line is actuals
# Prediction seems to underestimate the amplitude of the seasonal variance, with lower peak rainfall levels in winter, and higher rainfall lows in the summer




## 4 ARTIFICIAL NEURAL NETWORKS

# Use rain_matrix created earlier
# Create a Y dataset with one-month lag
X <- t(as.matrix(rain_matrix))
Y <- as.matrix(X[-1,])

# Training on first 80% of data (0.8*2232 = 1785.6 = 1786)
# Using same approach as for temp data used in class tutorial
# Doing regression so linout (linear output) is TRUE
rain.nnet <- nnet(X[1:1786, 1:10], Y[1:1786, 1:10], decay=1e-4, linout=TRUE, size=10)

# Take a look at the fitted values
rain.nnet["fitted.values"]
# For each region, all values seem to be constant or one of two fixed values within a narrow range - no seasonality
# Run the prediction on the remaining 20% of data (445 months)
# Y data set finishes at 2231 as there is one less month of data in Y compared to X
rain.pred <-predict(rain.nnet, Y[1787:2231, 1:10])


# Look at results for the first region: Scotland N
rain.pred

matplot(cbind(Y[1787:2231,1], rain.pred[,1]), ylab="Monthly average rainfall", xlab="Time (in months)",
main="Scotland N", type="l")
# Looks bad
# Possible explanations - wrong parameters - have tried many different combinations of decay and size, to no avail -
tried mygrid approach from tutorial but suspect that only works with regression rather than time-series
# Is dataset too large?


# Try nnetar() adapted from Hyndman and Athanasopoulos, as this can automatically select the number of lagged
inputs and the number of neurons in the hidden layer
# nnetar() will also add a seasonal lag component if seasonality is identified
# set lambda to zero to force positive values (cannot have -ve rainfall!)
# Use Scotland N regional data
rain.fit <- nnetar(rain_matrix[1,], lambda=0)  # this takes quite a long time
autoplot(forecast(rain.fit, h=30))
# This uses a NNAR(29,15) model
# Looks better than nnet() results above, but dataset too long to really see, so try with shorter timeseries

# Using a reduced timeseries of 360 months for Scotland N
rain.fit <- nnetar(rain_matrix[1,1873:2232], lambda=0)
autoplot(forecast(rain.fit, h=50))
# Uses an NNAR(18,10) model and results look encouraging
# Amplitude of seasonal variance possible underestimated compared to historic actuals
# This forecasts out 50 months into the future from end of timeseries, so not able to compare prediction with actuals


## END OF CODE ##
```