DAT Word 2315053

2024-05-28

# ABBREVIATIONS

ST - Statistical Analysis AI - Artificial Intelligence MLR - Multiple Linear Regression ML - Machine Learning TSF - Time Series Forecasting RNN - Recurrent Neural Network DA - Data Analysis WF - Weather Forecast SVM - Support Vector Machine RNN - Recurrent Neural Network. MLR - Multiple Linear Regression RFR - Random Forest Regression TSAF - Time Series Analysis and Forecasting MAE - Mean Absolute Error RMSE - Root Mean Squared Error CRISP-DM - Cross-Industry Standard Process for Data Mining MAE - Mean Absolute Error ARIMA - Autoregressive Integrated Moving Average

# ABSTRACT

This study explores the use of three different machine learning models, Multiple Linear Regression (MLR), Support Vector Regression (SVR), and Random Forest Regression (RFR), for weather forecasting in the Scarcliffe region. This project will document the preprocessing procedures used to clean the unprocessed weather data before the models are trained and tested. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics are used to assess the performance and accuracy of the model.

The project, which is written in the R programming language, uses several libraries for data manipulation, visualisation, and analysis, including zoo, DescTools, ggplot2, tidyverse, dplyr, and naniar. The goal of the study is to produce insightful information about weather forecasting that will be useful to a variety of stakeholders, such as residents, farmers, tourism operators, flight navigation services, and policymakers.

This research compares the performance of the MLR, SVR, and RFR models to determine which method is best for efficient weather prediction in the Scarcliffe region. It is anticipated that the results would improve weather forecasts’ accuracy and usefulness, which will help with resource management and well-informed decision-making in a variety of industries.

# 1.0. INTRODUCTION

As climatic variability becomes more visible, understanding its implications on agricultural production and tourism is critical to ensuring economic stability and growth. This research aims to examine local weather patterns and their effects on agricultural and tourism sector in Scarcliffe, UK. The aim of this research is to offer a thorough analysis, relevant correlations, trends, and patterns by assessing the direct relationships between certain weather parameters and economic outcomes, agricultural production and tourism flows in Scarcliffe. It will equip stakeholders, particularly local farmers, tourism providers, and residents, with adequate data-driven insights that will help boost their operations and strategic objectives. Accurate weather forecasting plays a significant role in practically every sector and society, as it allows businesses to make informed decisions, emergency personnel to prepare for natural tragedies and people to organise their activities (Weather Forecast Models: The Ultimate Guide (2024), no date). Pliske et al. (2004) discussed that technological advancement have greatly influenced the technologies accessible to weather forecasters in recent years. Today’s forecasters are frequently overwhelmed by the enormous variety of technological products, the development of these tools has made it simple for us to conduct research on weather in various locations for the benefit of many stakeholders, including businesses, farmers, and people. However, in this study, I will be primarily focusing on Scarcliffe data to understand how different climate variables relates with one another.

## 1.1. ABOUT THE LOCATION

Scarcliffe offers an interesting case study for investigation as it has a population of 5,288 in the UK Census of 2011 (‘Scarcliffe’, 2024). It is situated 127 miles north-west of London, two miles south-east of Bolsover, and six miles north-west of Mansfield. Two miles to the northwest of Nottinghamshire’s boundary is Scarcliffe. Scarcliffe could be found in latitude 53.227°N and longitude -1.224°E. The town is within the jurisdiction of the Bolsover District Council, which is part of the Derbyshire County Council (Where is Scarcliffe? Scarcliffe on a map, no date). Gaining a comprehensive understanding of the weather patterns in Scarcliffe is of utmost importance for residents of the town, local farmers, tourist providers, meteorologists, and policymakers. This project aims to utilise a comprehensive dataset containing weather parameters for Scarcliffe to address various research questions and provide insights into local weather patterns.

## 1.2. SIGNIFICANCE OF SCARCLIFFE LOCATION

Scarcliffe’s geographical position in the East Midlands makes it vulnerable to a wide range of weather patterns, which are impacted by its closeness to the Pennines and the North Sea. Its weather impacts various sectors, including agriculture, tourism, and local infrastructure.

## 1.3. PROBLEM STATEMENT

Scarcliffe is a significant location, but its weather patterns have not been thoroughly studied, which makes it difficult to forecast and prepare for bad weather. To close this gap and give stakeholders useful information, a thorough examination of past weather data is required. A closer analysis of Scarcliffe’s weather data is needed to improve prediction accuracy. This would lead to better agricultural outcomes as well as enhanced ecological management and preparedness for emergencies. The potential effects of global climate change make it essential to update and enhance weather prediction models often in order to maintain their relevance and usefulness. In order to address these issues, more precise and locally applicable weather forecasting models will be developed. Sophisticated statistical and machine learning techniques will be employed, together with comprehensive data analysis, to provide a deeper knowledge that will enable Scarcliffe residents, local farmers, and businesses to make informed decisions.

## 1.4. SIGNIFICANCE OF DATA ANALYSIS

### 1.4.1. Rationale for Data Analysis

The basis for analysing Scarcliffe’s weather prediction data comes from the requirement to precisely comprehend and predict environmental conditions. This analysis is critical for various reasons. • Enhanced Predictive Power: By analysing previous weather data, predictive models could be created to better forecast future weather conditions. This is critical for making decisions and planning across several industries. • Risk Management: DA assists in finding patterns and trends that indicate future hazardous weather occurrences. This knowledge enables improved preparedness and risk reduction. • Resource Allocation: Effective DA provides effective resource allocation by forecasting weather-dependent occurrences, which may have a substantial impact on both the economy and the environment.

### 1.4.2. Who are the Stakeholder?

• Farmers and Agricultural Sector: Accurate weather forecasts are essential for farmers and other agricultural workers to plan planting, watering, and harvesting. Predictive weather analysis reduces the likelihood of crop failure while assisting in output growth. • Local governments and emergency management agencies: They depend on accurate weather forecasts to help them anticipate and react to extreme weather events, ensure public safety, and reduce financial losses. • Tourism Industry: The weather has a significant effect on tourism. Precise weather data is helpful to Scarcliffe’s tourism sector regarding marketing and management of operations. • Environmental Agencies: The study can support further attempts to track and comprehend how local weather patterns are affected by climate change.

### 1.4.3. Impact on Specific Organisation Decision Making

• Transportation: Weather trends and forecasts have an impact on transportation time management, route planning, and safety protocols. This helps to minimise weather-related hazards and ensure on-time delivery. • Building & Infrastructure: Planning time frames for maintenance and allocation of resources for building projects, as well as designing and building resilient infrastructure, may all be facilitated by an understanding of local weather dynamics. • Farmer’s Community: Fluctuations in temperature, precipitation, and extreme weather phenomena can all influence the duration of growing seasons, the amount of crops produced, and the existing agricultural methods. • Tourist Operators: The tourist industry depends significantly on reliable weather for activities and attractions, is subject to uncertainties that may have an impact on visitor numbers and earnings.

# 2.0. LITERATURE REVIEW

ML models for WF has gained a lot of significance in recent years because of their capacity to enhance prediction accuracy and reliability. Several researchers have investigated the use of different methods of ML for WF, drawing on historical weather data and meteorological variables. Previous research has demonstrated the value of ML, time-series modelling, and statistical analysis when examining weather data. many ML models including random forests and neural networks, as well as statistical models, such as SARIMA and ARIMA, have been studied for WF. However, studies on Scarcliffe’s weather patterns have been limited, providing an opportunity for further investigation. In one study, Mohandes et al. (2004) developed a reliable forecasting model by applying ensemble learning techniques including gradient boosting machines and random forests. Their research made clear how crucial it is to integrate many data sources and use feature engineering to improve wind speed prediction. Singh et al. (2019) also utilised three ML models, ANN, RNN, and SVM for weather prediction in one of their research. Lagerquist et al. (2019), on the other hand, tested multiple machine learning algorithms, including logistic regression, decision trees, RFR, neural networks and gradient-boosted forests, for forecasting straight-line wind alone.

# 3.0. METHODOLOGY

The project will employ the CRISP-DM methodology, it encompasses these subsequent stages:

## 3.1. BUSINESS UNDERSTANDING:

### 3.1.1. Objectives

1. Understand the correlations and patterns between weather parameters in the Scarcliffe dataset.
2. Create statistical models that measure correlations among variables and test hypotheses.
3. Create time-series forecasting models to anticipate future weather conditions in Scarcliffe.
4. Deploy ML regression approaches to identify complicated, non-linear correlations between weather parameters and provide Scarcliffe with accurate weather predictions.
5. Use appropriate assessment metrics and cross-validation procedures to check the performance and accuracy of the produced statistical models, SVR, and ML predictions.

### 3.1.2. Research Questions

1. Perform normality tests (e.g., Shapiro-Wilk, Kolmogorov-Smirnov) to evaluate whether the RAINC data follows a normal distribution or deviates from it.
2. Is there a substantial correlation between Surface Temperature (TSK) and Soil Temperature (TSLB) at Scarcliffe?
3. Are there any underlying patterns or clusters in the Scarcliffe weather data that can be detected with multivariate analysis techniques such as principal component analysis (PCA) or cluster analysis?
4. How do the associations between weather parameters in the Scarcliffe dataset change over time or seasons, and can statistical or machine learning models account for these variations?
5. Can machine learning regression models capture the complex relationship between meteorological variables and produce accurate predictions for temperature, pressure, humidity, and wind speed at Scarcliffe?

## 3.2. DATA UNDERSTANDING

### 3.2.1. About Dataset

The data chosen for this assessment talks about weather research and Forecasting in May 2018 of various regions from which I extracted my Scarcliffe, UK dataset for my analysis. The dataset raw dataset was downloaded from my Data Science course Moodle: URL: (<https://moodle.bolton.ac.uk/mod/resource/view.php?id=2241359>)

There are 5,451 rows and 2,482 columns in the raw dataset, notable columns are: XLAT, XLONG, TSK, PSFC, U10, V10, Q2, RAINC, RAINNC, SNOW, TSLB, SMOIS. The remaining columns are repetition of the above listed columns. First row contains the spread out of the time and date. I will be showing step by step of how I reconstruct the whole dataset, and how I got my specific location: Scarcliffe. I made use of google map to check the XLAT and XLONG of various location before getting my Scarcliffe location, XLAT = 53,227 and XLONG = -1,224

These are the description and measuring unit of the weather parameters in the dataset.

##### Table 1: Description and measuring unit of the weather parameters.

##### Figure 1: Snippet of the raw Weather Research & Forecasting Dataset

Source: Course Moodle (<https://moodle.bolton.ac.uk/mod/resource/view.php?id=2241359>)

## 3.3. DATA PREPARATION:

Step by step documentation and visualisation of how we transformed the raw data into meaningful data suitable for our analysis will be done here. During this phase, the data is cleaned, transformed, and pre-processed as needed before being analysed. Handling missing values, encoding categorical variables, scaling features, and separating the data into training and testing sets are some of the possible tasks. I carried out all my data preprocessing and analysis on RStudio, an integrated Development Environment (IDE), It offers an intuitive interface that simplifies the writing, running, and debugging of R code.

## Brief:

Looking at the “Weather Research & Forecasting” dataset, I understand the first thing I need to do is to restructure the data as it was not well arranged at the point of download. Firstly, I used google map to get my preffered location using the Longitude and Latitude, my location is Scarcliffe, Derbyshire. The dataset contains total of 2,482 columns as explained earlier, my understanding of the dataset is that, we have just 12 major columns – XLAT, XLONG, TSK, PSFC, u10, v10, Q2, RAINC, RAINNC, SNOW, TSLB, and SMOIS, other columns are repition of this 10 columns excluding the XLAT and XLONG columns, which means we have to restructure it in a way that, we will break the whole 2,482 columns into list of groups with each group having 10 columns each, after which, we will bind them altogether. In the raw dataset, the first row is actually not our header, those are our time and date column which we have to restructure as a column, step by step on how I achieved this will be shown below.

### 3.3.1. Import the Data

Importing the downloaded raw data into R is the initial step in every data analysis job. We can use many functions to read the data into R depending on the format of the data. In this instance, the read.csv() function can be utilised as the downloaded file is in CSV format, we can use View() function, this will pop out in another table for us to view the values. To use R seamlessly, there are libraries and packagies we have to download some of which are tidyverse, dplyr, ggplot2, zoo, naniar, corrplot and so on.

getwd()

## [1] "C:/Users/user/OneDrive/Documents/DAT7006/ASS 2"

setwd("C:/Users/user/OneDrive/Documents/DAT7006/ASS 2")  
  
#Read the CSV file and skipp the first row  
Weather\_data <- read.csv("Weather Research & Forecasting Dataset 2018.csv", header = FALSE, skip = 1)

### 3.3.2. Explore the Data

It is very important to explore and comprehend the data’s variables, structure, and possible problems after it has been imported. To obtain a summary of the data, we can utilise functions like head() - this will show the 5 5 rows of the dataset, tail() - this will show last 5 rows of the datset, str() - this function will show the data type of all variable, this is very important as it gives us insight of the variable we are working on and to know if to change accordingly. The whole restructurig of my data was done in the process, I transposed the first row which has my date and time into a new column “Datetime” and other necessary steps follows like breaking the huge dataset into just 13 columns as explained in my brief.

#we set the column names as character and skipped first rol of the Weather\_data  
colnames(Weather\_data) <- as.character(unlist(Weather\_data[1, ]))  
Weather\_data <- Weather\_data[-1, ]  
  
  
View(Weather\_data)  
  
#made a subset where XLAT == 53.227 & XLONG == -1.224  
Scarcliffe\_data <- subset(Weather\_data, XLAT == 53.227 & XLONG == -1.224)  
View(Scarcliffe\_data)  
  
#I removed the first and second columns (XLAT & XLONG) in order to be able to split   
#the columns to groups of 10 each as the first 10 columns are iterated all over for   
#the remaining columns.  
  
ScarcliffeXX <- Scarcliffe\_data[, -(1:2)]  
  
View(ScarcliffeXX)  
  
#Splitted Mansfield\_columns dataframe into groups with 10 columns each  
Scarcliffe\_columns <- split.default(ScarcliffeXX, rep(1:248, each = 10))  
  
  
View(Scarcliffe\_columns)  
  
#used lapply function combine the XLAT & XLONG columns back and naming each columns  
Scarcliffe\_Join <- lapply(Scarcliffe\_columns, function(x) {  
 Scarcliffe\_bind <- cbind(Scarcliffe\_data[, 1:2], x)  
 colnames(Scarcliffe\_bind) <- c("XLAT", "XLONG", "TSK", "PSFC", "u10", "v10", "Q2", "RAINC", "RAINNC", "SNOW", "TSLB", "SMOIS")  
   
 return(Scarcliffe\_bind)  
})  
  
View(Scarcliffe\_Join)  
  
#combine the lists of data frames into one single data frame  
Scarcliffemain\_df <- do.call(rbind, Scarcliffe\_Join)  
  
  
### After restructing the dataset, I will be creating new variable "Datetime" using the time as.POSIXlt  
  
#Let's generate the sequence of time stamps with 3-hour intervals  
# Define the start date and time using as.POSIXlt  
start\_date <- as.POSIXlt("01-05-2018 00:00:00", format = "%d-%m-%Y %H:%M:%S")  
  
nrows <- nrow(Scarcliffemain\_df)  
View(nrows)  
  
# Create a sequence of timestamps with a 3-hour interval for additional rows  
timestamp\_sequence <- seq(start\_date, by = "3 hours", length.out = nrows)  
  
# extract the format, how we want it to be arranged  
timestamp <- format(timestamp\_sequence, "%d-%m-%Y %H:%M:%S")  
  
Scarcliffemain\_df$Datetime <- timestamp  
  
  
# View the resulting data frame  
View(Scarcliffemain\_df)  
head(Scarcliffemain\_df)

## XLAT XLONG TSK PSFC u10 v10 Q2 RAINC RAINNC SNOW TSLB SMOIS  
## 1 53.227 -1.224 272.1 99473 3.4 -1.8 0.00335 0.0 0.0 0.0 278.6 0.2901  
## 2 53.227 -1.224 273.2 99452 4.4 -0.8 0.00389 0.0 0.0 0.0 277.5 0.2896  
## 3 53.227 -1.224 278.0 99468 5.2 -0.3 0.00375 0.0 0 0.0 276.8 0.2891  
## 4 53.227 -1.224 287.7 99446 4.0 3.4 0.00384 0.0 0.0 0.0 277.0 0.2874  
## 5 53.227 -1.224 291.9 99359 4.6 4.9 0.00504 0.0 0.0 0.0 281.0 0.2864  
## 6 53.227 -1.224 288.1 99259 6.4 5.7 0.00494 0.0 0.0 0.0 283.6 0.2849  
## Datetime  
## 1 01-05-2018 00:00:00  
## 2 01-05-2018 03:00:00  
## 3 01-05-2018 06:00:00  
## 4 01-05-2018 09:00:00  
## 5 01-05-2018 12:00:00  
## 6 01-05-2018 15:00:00

tail(Scarcliffemain\_df)

## XLAT XLONG TSK PSFC u10 v10 Q2 RAINC RAINNC SNOW TSLB SMOIS  
## 243 53.227 -1.224 288.2 100267 -1.4 0.3 0.00962 0.0 0.0 0 285.2 0.2553  
## 244 53.227 -1.224 291.9 100358 -1.6 -0.3 0.01049 0.0 0.0 0.0 286.5 0.2549  
## 245 53.227 -1.224 303.6 100317 -2.2 -0.2 0.01147 0.0 0.3 0.0 289.9 0.2533  
## 246 53.227 -1.224 298.9 100253 -4.2 -0.4 0.01204 1.3 0.3 0.0 292.3 0.2556  
## 247 53.227 -1.224 294.8 100244 -5.4 0.4 0.01152 1.3 0.3 0.0 293.1 0.2540  
## 248 53.227 -1.224 286.6 100329 -2.8 -0.4 0.01006 1.3 0.3 0.0 291.7 0.2536  
## Datetime  
## 243 31-05-2018 06:00:00  
## 244 31-05-2018 09:00:00  
## 245 31-05-2018 12:00:00  
## 246 31-05-2018 15:00:00  
## 247 31-05-2018 18:00:00  
## 248 31-05-2018 21:00:00

# The data types are all in character form, I need to change all back to there proper data type  
  
#data types - change each column to the right data type  
  
Scarcliffemain\_df <- Scarcliffemain\_df %>%  
 mutate\_at(1:12, as.numeric)  
View(Scarcliffemain\_df)  
  
#convert the Datetime column to date format  
Scarcliffemain\_df$Datetime <- as.POSIXlt(Scarcliffemain\_df$Datetime, format = "%d-%m-%Y %H:%M:%S")  
  
# Now, recheck to be sure they are in right format now  
str(Scarcliffemain\_df)

## 'data.frame': 248 obs. of 13 variables:  
## $ XLAT : num 53.2 53.2 53.2 53.2 53.2 ...  
## $ XLONG : num -1.22 -1.22 -1.22 -1.22 -1.22 ...  
## $ TSK : num 272 273 278 288 292 ...  
## $ PSFC : num 99473 99452 99468 99446 99359 ...  
## $ u10 : num 3.4 4.4 5.2 4 4.6 6.4 1.9 1.3 1.1 0.5 ...  
## $ v10 : num -1.8 -0.8 -0.3 3.4 4.9 5.7 5.4 7.4 8.1 9.2 ...  
## $ Q2 : num 0.00335 0.00389 0.00375 0.00384 0.00504 0.00494 0.00532 0.00492 0.00509 0.00544 ...  
## $ RAINC : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ RAINNC : num 0 0 0 0 0 0 0 0 0 0.3 ...  
## $ SNOW : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ TSLB : num 279 278 277 277 281 ...  
## $ SMOIS : num 0.29 0.29 0.289 0.287 0.286 ...  
## $ Datetime: POSIXlt, format: "2018-05-01 00:00:00" "2018-05-01 03:00:00" ...

# then view  
View(Scarcliffemain\_df)

### 3.3.3. Handling Missing Values

As stated by (Horton and Lipsitz, 2001) and corroborated by (Yadav and Roychoudhury, 2018), Missing data typically affects data analysis in scientific research. In recent decades, there has been a lot of study on statistical strategies for dealing with missing data. Missing values are common in datasets, and they must be handled carefully. Missing values can be identified using functions such as is.na() and sum(is.na()). Depending on the analysis needs and the nature of the missing data, we can either eliminate the rows or columns with missing values or impute them using appropriate procedures. In this project, I made use of approximation (na.approx) to handle my missing values, using mean or dropping the NA cells is not ideal for my analysis. After thorough restructuring of my dataset, I realised we I have total 67 missing values.

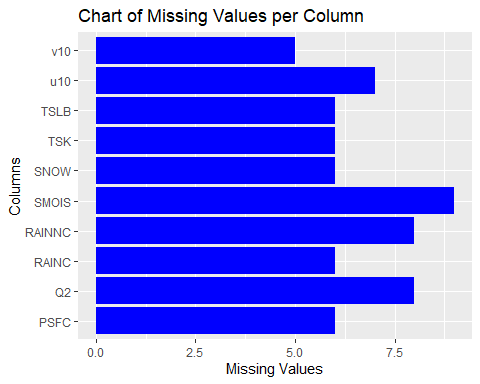
# Checking for missing values and handling them  
  
#gives us breakdown of NA in each columns  
colSums(is.na(Scarcliffemain\_df))

## XLAT XLONG TSK PSFC u10 v10 Q2 RAINC   
## 0 0 6 6 7 5 8 6   
## RAINNC SNOW TSLB SMOIS Datetime   
## 8 6 6 9 0

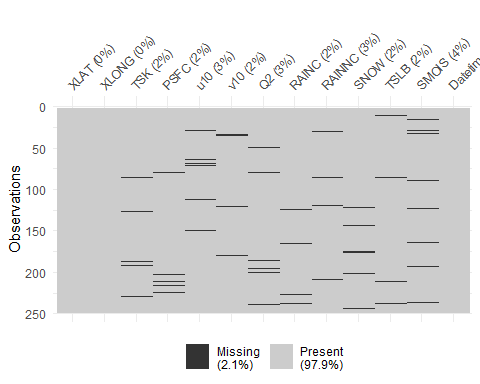
#I will create a dataframe of the na\_values  
missing\_values <- data.frame(  
 Columns = c("XLAT", "XLONG", "TSK", "PSFC", "u10", "v10", "Q2", "RAINC", "RAINNC", "SNOW", "TSLB", "SMOIS", "Datetime"),  
 na\_count = c(0, 0, 6, 6, 7, 5, 8, 6, 8, 6, 6, 9, 0)  
)  
View(missing\_values)  
  
# I filter out rows with 0 NA\_count as they won't be needed  
missing\_values1 <- missing\_values[missing\_values$na\_count > 0, ]  
missing\_values1

## Columns na\_count  
## 3 TSK 6  
## 4 PSFC 6  
## 5 u10 7  
## 6 v10 5  
## 7 Q2 8  
## 8 RAINC 6  
## 9 RAINNC 8  
## 10 SNOW 6  
## 11 TSLB 6  
## 12 SMOIS 9

#view filtered dataframe after removing rows/values with 0  
View(missing\_values1)  
  
# Visualisation of the missing values  
  
# The bar chart displays the missing values in each columns  
# The second plot shows the percent of missing values in each column  
  
#Below is the plot for our missing values on each column  
  
ggplot(data = missing\_values1, aes(x = na\_count, y = Columns)) +   
 geom\_bar(stat = "identity", fill = "blue") +  
 labs(title = "Chart of Missing Values per Column",   
 x = "Missing Values",   
 y = "Columns")



#this will show us the percentage of missing values in each column  
vis\_miss(Scarcliffemain\_df)



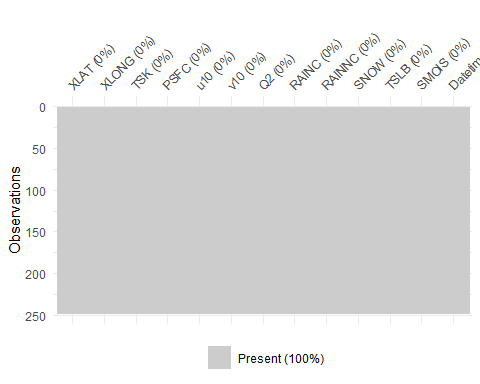
View(Scarcliffemain\_df) # this gives us overview of all our missing values  
  
  
# Handling of missing values, na.approx function was called to action here to handle the NA values  
  
# Let's handle all missing values column by column  
colSums(is.na(Scarcliffemain\_df))

## XLAT XLONG TSK PSFC u10 v10 Q2 RAINC   
## 0 0 6 6 7 5 8 6   
## RAINNC SNOW TSLB SMOIS Datetime   
## 8 6 6 9 0

# na.approx function will be used to interpolate NA values in each column  
Scarcliffemain\_df1 <- Scarcliffemain\_df %>%  
 mutate(across(c(TSK, PSFC, u10, v10, Q2, RAINC, RAINNC, SNOW, TSLB, SMOIS), na.approx))  
  
View(Scarcliffemain\_df1)  
  
# Let's recheck to be sure we don't have missing values anymore  
# All missing values are now handled, which means we have no missing value anymore  
colSums(is.na(Scarcliffemain\_df1))

## XLAT XLONG TSK PSFC u10 v10 Q2 RAINC   
## 0 0 0 0 0 0 0 0   
## RAINNC SNOW TSLB SMOIS Datetime   
## 0 0 0 0 0

#let's recheck our missing values percentage again  
vis\_miss(Scarcliffemain\_df1)



#the visualisation above shows that we don't have any missing value anymore