Maximum Temperatures in Britain in 2020

Contents

Introduction

The objective of this report is to perform an analysis of the maximum temperatures in Britain in the year 2020 and summarise the spatial and temporal variations that occur. We have two datasets that are taken from the UK Met Office, one which contains the name and location of the regions as well as the other dataset that has the maximum daily temperatures of the regions in the year 2020. On assessment of the data we find that there are four key locations in which the analysis must be done. The primary question that we try to answer is to fit a spatial model to predict the highest temperature in Morecambe, Coventry and Kinross on the 12th of September 2020. The second question which we must do is to fit a time series model to predict the maximum temperature in Yeovilton between the 1st and 7th of November 2020 and asses the model's appropriateness with respect to other locations. After this is complete we also extend the models to better understand the spatial and temporal variability of the maximum temperatures in Britain in 2020. We will also include a map of the predicted temperatures and also a map of the uncertainties on a 0.1 degree grid over Britain.

Initial Data Analysis

The first step that we undertake is to understand the data and the already existing maximum temperatures of the regions found. Based on this evidence when we predict our maximum temperatures using spatial and time series models, we will have a rough estimate as to whether the predictions are suitable or are abnormally high or low. Since the four regions we take into consideration for prediction are shown it is valid to note that the maximum temperatures of Morecambe is 31.1, Coventry is 35.1, Kinross is and for Yeovilton it is 33.1. Let us also note that the actual maximum temperature on September 12th, 2020 for Morecambe, Coventry and Kinross was **17.3**, **19.6** and **15** respectively. The next step we undertake is that we have two different datasets which hold valuable information for this analysis and thus we merge them together. So now we have one dataset which contains information of the location, date, latitudinal and longitudinal

coordinates as well as the elevation and max temperature. We can see from the figure(1) below there seems to a few outliers in Morecambe and Coventry but none in Kinross. We see that there is

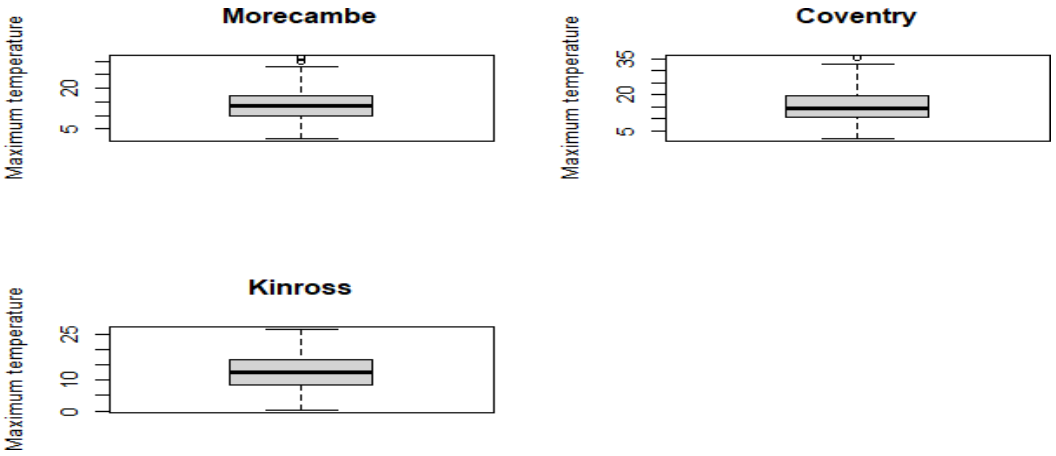

Figure 1: Boxplots for Morecambe, Coventry, and Kinross with outliers.

This gives us an idea of the outliers now let us look at the graphical and numerical summaries of the data.

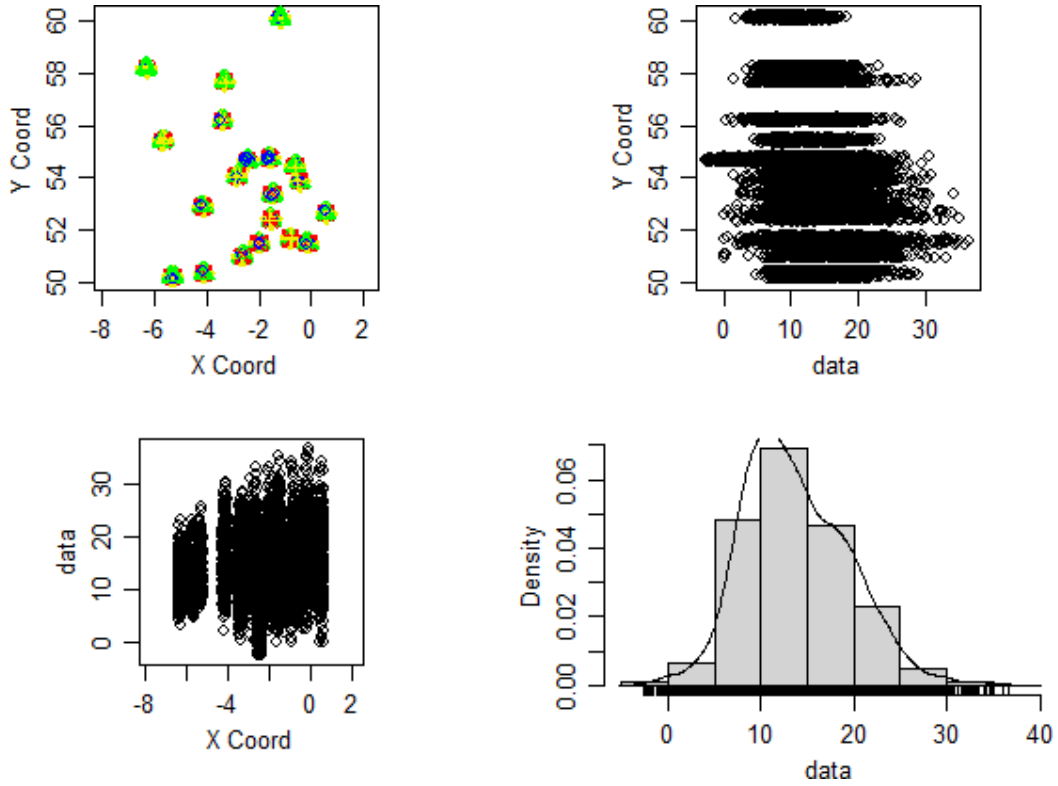

Figure 2: Graphical summary of maximum temperatures across the UK.

We can also see that there is a pretty large dataset with 366 datapoints for Morecambe, Coventry and Kinross so the numerical summary is shown below

Number of data points: 366

Coordinates summary

	Longitude	Latitude
min	-3.504464	52.32602
max	-1.440880	56.30899

Distance summary

min	max
0.0003256793	4.4276626742

Data summary

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.60000	10.10000	13.80000	14.33142	18.07500	32.80000

We can see from figure(2) as well as from the numerical summary that as the latitude increases which means the further north the maximum temperature tends to be lower which indicates a spatial pattern and thus the generation of spatial and temporal models is justified in this case.

Methods

There are two main methods that we are going to implement on this project, they are spatial and temporal models. Let us initially delve into spatial model methodology.

Spatial Method

We are using this technique to predict the maximum temperature in Morecambe, Coventry and Kinross on the 12th of September 2020. We use ordinary kriging to help determine an accurate prediction of the maximum temperature. Ordinary kriging is the most widely used kriging method. It serves to estimate a value at a point of a region for which a variogram is known, using data in the neighbourhood of the estimation location (Wackernagel, H. 2003). but before we do that we must first determine if the variables are isotropic or not. We must also make the assumption that the data is stationary. Using the geoR library and the variog function we are able to lot the variograms. Then we fit the Mattern models with different kappa values on them to determine which is the best fit, this cannot be done visually as they look similar in nature and thus we use the sum of squares to determine which is a better fit. This is relatively simple as the model with the lesser or lower sum of squares tends to be the

better fitted model. After this we predict our values using the ordinary krige by extracting the coordinates of each point using the krige.conv function that will generate the predicted value as well as the variance of each location. Finally, we validate the models to determine if it makes sense to use this model or not. We can see the variogram shown below with respect to the locations on September 12th 2020.

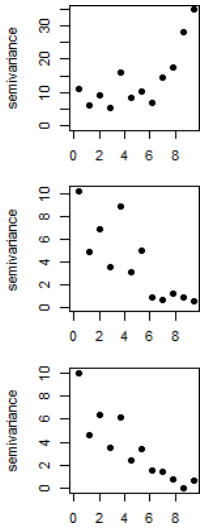

Figure 7: Variogram of Morecambe, Coventry and Kinross.

Temporal Method

The location we use is that of Yeovilton on the dates November 1st to 7th 2020 in order to predict the maximum temperature. The first step of this process tends to be converting the data to a timeseries dataset which makes the generation the models much easier. We also check to see if there are any seasonal trends which might affect the fitting of an appropriate model. In the case the data is stationary it is best we find the first and second order and use that to continue the prediction process. We then check the ACF and PACF plots to check if the MA, ARMA or ARIMA model is best suited to this dataset. The auto arima model is used to give us the best suited estimators. The second model we fit is a DLM model. The main features of the package are its flexibility to deal with a variety of constant or time-varying, univariate or multivariate models, and the numerically stable singular value decomposition-based algorithms used for filtering and smoothing. (Petris, G. 2010). Let us have a look at the distribution of max temperature across Yeovilton over time to determine if it stationary or not.

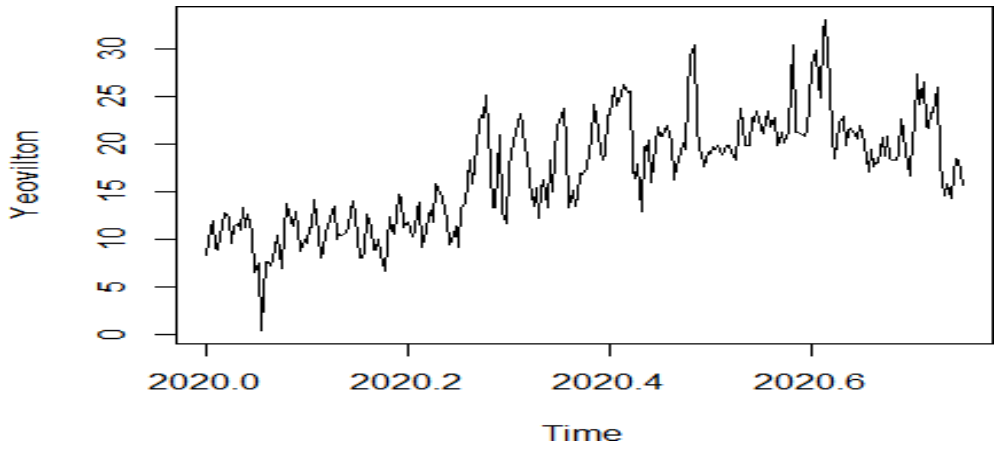

Figure 3: Maximum temperature in Yeovilton over a period of time.

There seems to be a trend in the maximum temperature and this leads us to believe that the data isn't stationary.

Results

Spatial Model

We plot the required variogram of the locattions on the given date as seen in figure 4, We then fit a mattern model with kappa 0.5 which is the default and also kappa 1, and we see that the sum of squares is lesser in kappa 1 so we chose that mattern model to predict using ordinary krige and we find the following figure 7

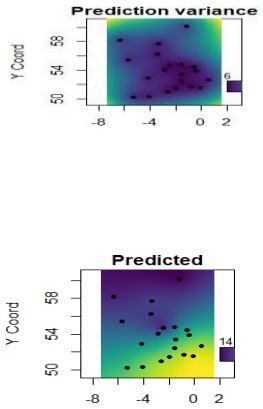

Figure 7: Predicted mean and Variance.

Then we validated the model and got the following results.

xvalid: number of data locations = 20

xvalid: number of validation locations = 20

xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,

xvalid: end of cross-validation

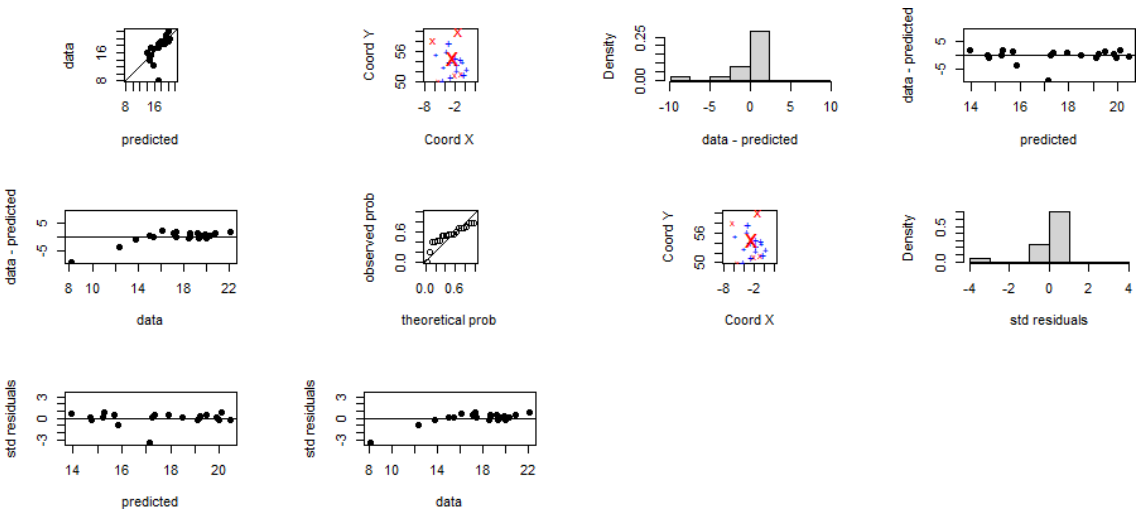

Figure 8: Validation Graphs.

Temporal Model

Based on the objective of this project which is to predict the maximum temperature in Yeovilton from November 1st -7th 2020, we do so by initially finding the first and second order difference and then plotting the ACF and PACF to determine if the model is better suited to be an MA, ARMA or ARIMA model. Look at the figure below

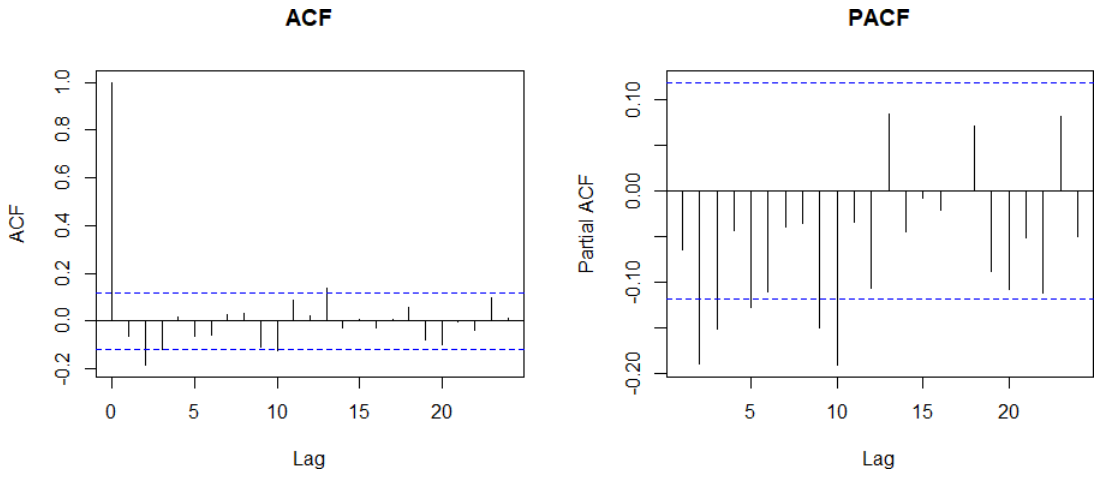

Figure 4: ACF and PACF of the Yeovilton timeseries data .

We can see that the ACF and the PACF both tend to move towards 0 and thus we arent't able to determine which is the best suited model so we use the auto arima function which determines an ARIMA(2,1,1) model. Since this model is a good fit and has statistically significant ar1, ar2 and ma1 values as well as a low AIC, we go ahead with the ARIMA(2,1,1) model. We then use this ARIMA(2,1,1) model to predict the values in Yeovilton from November 1st -7th 2020. For the predictions of Yeovilton from the 1st of November to 7th 2020, we find the predicted values to be 18.86015 across all 7 days with a variance of (4.545320, 4.571808, 4.598143, 4.624329, 4.650367, 4.676260, 4.702010).

Now we validate the model by plotting the residual diagnostics on the ARIMA(2,1,1) model that we have chosen as shown below in figure (5). Here we see that the P values are clearly above 0.05 which means that the model is a reasonable fit. So we use the same model to determine the validity in other locations

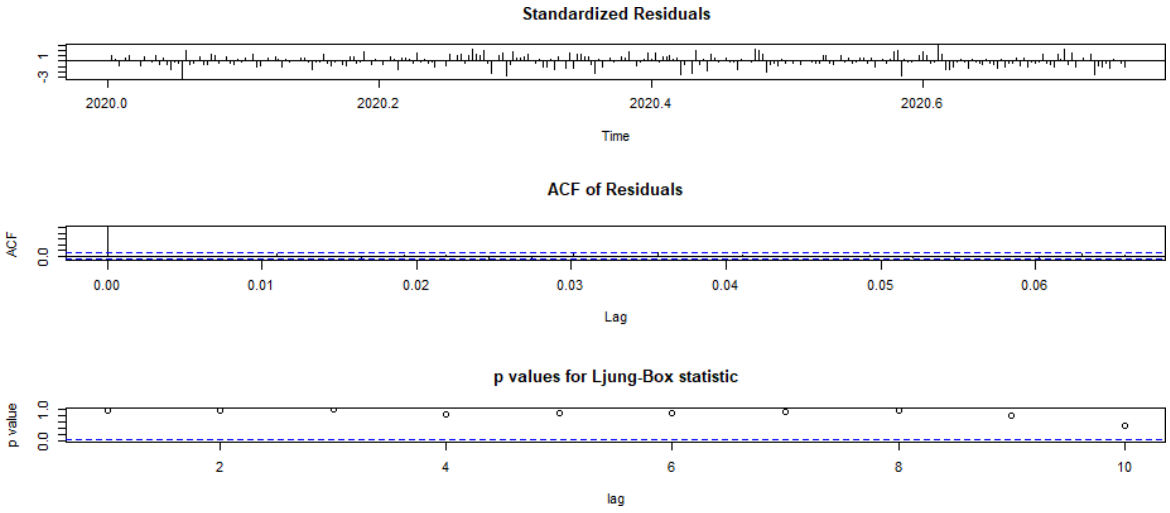

Figure 5: Residual diagnostics figure.

Location p-value

1 Machrihanish 0.297182964198248

2 High_Wycombe 0.827174070914029

3 Camborne 0.772949821411087

4 Dun_Fell 0.944405627335235

5 Plymouth 0.722200541764142

6 Durham 0.791103826253292

7 London 0.542924942400753

8 Porthmadog 0.0705528130850037

9 Morecambe 0.306836435726007

10 Kinross 0.814733135706819

11 Morecambe 0.980433689300864

12 Lossiemouth 0.722657544674664

13 Marham 0.608574052003374

14 Whitby 0.253527802812133

16 Yeovilton 0.488786404424921

17 Sheffield 0.469768668532784

18 Coventry 0.15388550056543

19 Stornoway 0.0325162336785436

20 Lyneham 0.459810484587804

We can observe that the p values for all the locations except 1 has a value greater than 0.05 which means that this model is well suited for almost all the locations.

Summary

In the spatial model we predict the max temp of 3 locations on 12th September using variogram and krige function then we checked which is the best fit model followed by predictions and validation of said model.

In the temporal method used to predict the maximum temperature of Yeovilton from 1 to 7 November, we first differentiate the original data and then use the auto arima function to get an ARIMA model which is then used to predict the values and is validated using the p values as well as the residual diagnostic figures. We then use this model on the other locations as well to determine that it is well suited to all of them.

Biblography

Petris, G. (2010). An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1–16. https://doi.org/10.18637/jss.v036.i12

Wackernagel, H. (2003). Ordinary Kriging. In: Multivariate Geostatistics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-05294-5_11

Appendix

#Time series part

```
Yd <- TempData[c('Date', Yeovilton)][TempData['Date']<20201001,]
#transfer the data into time series data
Yeovilton.ts <- ts(Yd['Morecambe'],start=c(2020,1,1),frequency =
365)plot(Morecambe.ts)
par(mfrow=c(1,2)) df1 Yeovilton
df2Yeoviltonts=diff(diff(Yeovilton.ts))
plot(df1Yeoviltonts, main='1st difference')
plot(df2Yeoviltonts, main = '2nd difference')

Yeoviltonts <- ts(Yd['Yeovilton'],start=c(2020,1,1))
df1Yeoviltonts=diff(Yeoviltonts)
df2Yeoviltonts=diff(diff(Yeoviltonts))
par(mfrow=c(1,2))
acf(df1Yeoviltonts, main = 'ACF'); pacf(df1Yeoviltonts, main = 'PACF')
maa <- auto.arima(Yeovilton.ts, max.p = 3, max.q = 3, max.d = 2)
maa
forecast <- predict(maa, n.ahead = 7)
forecast$pred #expectation
forecast$se #variance
valofp<-LocData['Location']
valofp['p-value']<-LocData['Location']
valofp <- valofp[which(valofp$Location!='Yeovilton'),]
other_loc <- LocData[which(LocData$Location!='Yeovilton'),]
for(i in 1:nrow(other_loc)){
temploc <- TempData[c('Date',other_loc$Location[i])][TempData['Date']<20201001,]
```

```
temploc.ts <- ts(temploc[other_loc$Location[i]],start=c(2020,01,01),frequency = 365)

temploc.model = arima(temploc.ts, order = c(2,1,1))

valofp[which(valofp['Location']==other_loc$Location[i]),'p-value'] = Box.test(resid(temploc.model), type =}

valofp
```

#Spatial part

```
Yd <- TempData[c('Date','Yeovilton')][TempData['Date']<20201001,]

#transfer the data into time series data

Yeovilton.ts <- ts(Yd['Yeovilton'],start=c(2020,1,1),frequency = 365)

plot(Yeovilton.ts)

par(mfrow=c(1,2))

df1Yeoviltonts=diff(Yeovilton.ts)

df2Yeoviltonts=diff(diff(Yeovilton.ts))

plot(df1Yeoviltonts, main='1st difference')

plot(df2Yeoviltonts, main = '2nd difference')


Yeoviltonts <- ts(Yd['Yeovilton'],start=c(2020,1,1))

df1Yeoviltonts=diff(Yeoviltonts)

df2Yeoviltonts=diff(diff(Yeoviltonts))

par(mfrow=c(1,2))

acf(df1Yeoviltonts, main = 'ACF'); pacf(df1Yeoviltonts, main = 'PACF')

maa <- auto.arima(Yeovilton.ts, max.p = 3, max.q = 3, max.d = 2)

maa

forecast <- predict(maa, n.ahead = 7)

forecast$pred #expectation

forecast$se #variance

valofp<-LocData['Location']

valofp['p-value']<-LocData['Location']

valofp <- valofp[which(valofp$Location!='Yeovilton'),]

other_loc <- LocData[which(LocData$Location!='Yeovilton'),]

for(i in 1:nrow(other_loc)){
```

```
temploc <- TempData[c('Date',other_loc$Location[i])][TempData['Date']<20201001,]

temploc.ts <- ts(temploc[other_loc$Location[i]],start=c(2020,01,01),frequency = 365)

temploc.model = arima(temploc.ts, order = c(2,1,1))

valofp[which(valofp['Location']==other_loc$Location[i]),'p-value'] = Box.test(resid(temploc.model), type =}

valofp



LocData <- read.csv('metadata.csv')

TempData <- read.csv('MaxTemp.csv')

mydata = merge(LocData, stack(TempData[which(TempData$Date == '20200815'),])[-1,], by.x = 'Location', byunobserved_loc <- c('Kinross','Morecambe',Coventry)

observed_loc <- mydata[-which(mydata$Location %in% unobserved_loc),]

# transfer data into geodata

geodata1<- as.geodata(observed_loc,coords.col=2:3,data.col=5)

summary(geodata1) # numerical

points(geodata1)

plot(geodata1) #graphical

vario <- variog(geodata1, option='bin',estimator.type = "classical")

par(mar=c(4,4,2,2))

plot(vario, pch = 19)

par(mfrow=c(1,2))

plot(variog(geodata1, option='cloud'),pch = 19)

plot(variog(geodata1, option='bin',bin.cloud = TRUE), bin.cloud = TRUE)

par(mar=c(4,4,2,2), mfrow=c(1,3))

variomod <- variog(geodata1, option='bin',estimator.type = "modulus",max.dist=12,bin.cloud = TRUE)

plot(variog(geodata1, option='cloud',estimator.type = "modulus",max.dist=12,bin.cloud = TRUE), pch = 19)

plot(variomod,pch = 19)

plot(variomod, bin.cloud = TRUE)

vma <- variofit(variomod)

plot(variomod)
```

```
lines(vma)

variolinear <- variofit(variomod,cov.model = 'linear')

vma1.0 <- variofit(variomod,kappa=1.0, fix.kappa=TRUE)

plot(variomod)

lines(variolinear,col='red')
lines(vma1.0,col='orange')

lines(vma,col='green')

# sum of squares of the version with kappa=1

summary(vma1.0)$sum.of.squares

# sum of squares of the version with kappa=0.5

summary(vma)$sum.of.squares

coordinate=LocData[LocData$Location %in% unobserved_loc,][,2:3]

preds <- krige.conv(geodata1, loc=coordinate, krige=krige.control(obj.model=vma1.0))

for (i in 1:(length(unobserved_loc))){
```