Predict the Unpredictable

Monthly Lake Erie Water Levels 1921-1970
STAT 621 Time Series Analysis

Benjamin Hulet - bhh2, Yuqiu Yang - yy44

Introduction

Located in North America, lake Erie is the fourth largest lake of the five Great Lakes by surface area. Albeit, the shallowest and smallest by volume. Lake Erie was said to be called, 'Erige' (cat), by local Native American tribes, because of its unpredictable and sometimes violently dangerous nature.

We chose to explore the Lake Eerie water level series sourced from the Time Series Data Library for a few reasons. First, the series spans a relatively lengthy period of time, 1921-1970, and includes 600 monthly observations which should allow us to develop a model with strong predictive capabilities. Additionally, through our research we found that environmental series have strong seasonal and cyclical components that were of primary interest. We were interested in exploring the Fast Fourier Transform, to understand the methodology, and to gain experience using it within this context where we delve into the inherent correlation between water levels over time and utilize the relationship to "peek" into the future.

The Series

In order to illustrate the patterns, we first plotted the series. Within the series, we split our data into three sections: the first section contains 60% of the data, the second 20% for tuning model parameters and the final 20% "hold-out" for assessing our model's predictive capability.

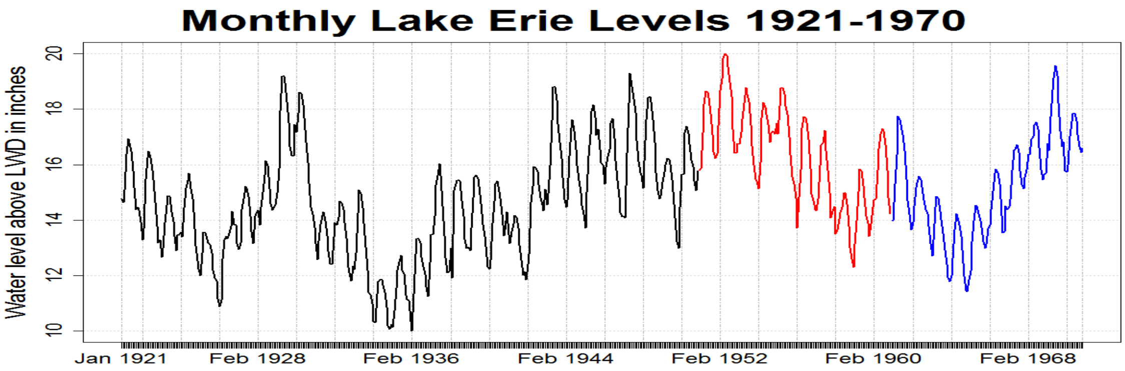


Figure 1: Monthly Lake Erie Water Levels split into three blocks

Looking at the first 60% of the series, other than the sharp rise in water level in 1928 and subsequent drought in the 1930's, the variance of the water level seems constant. Furthermore, we can see that there seems to be both a seasonal and cyclical component to the data. To confirm our observations, we evaluated the Acf and Pacf plots.

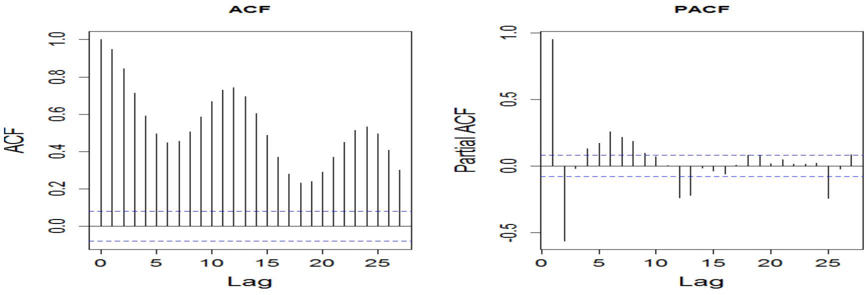
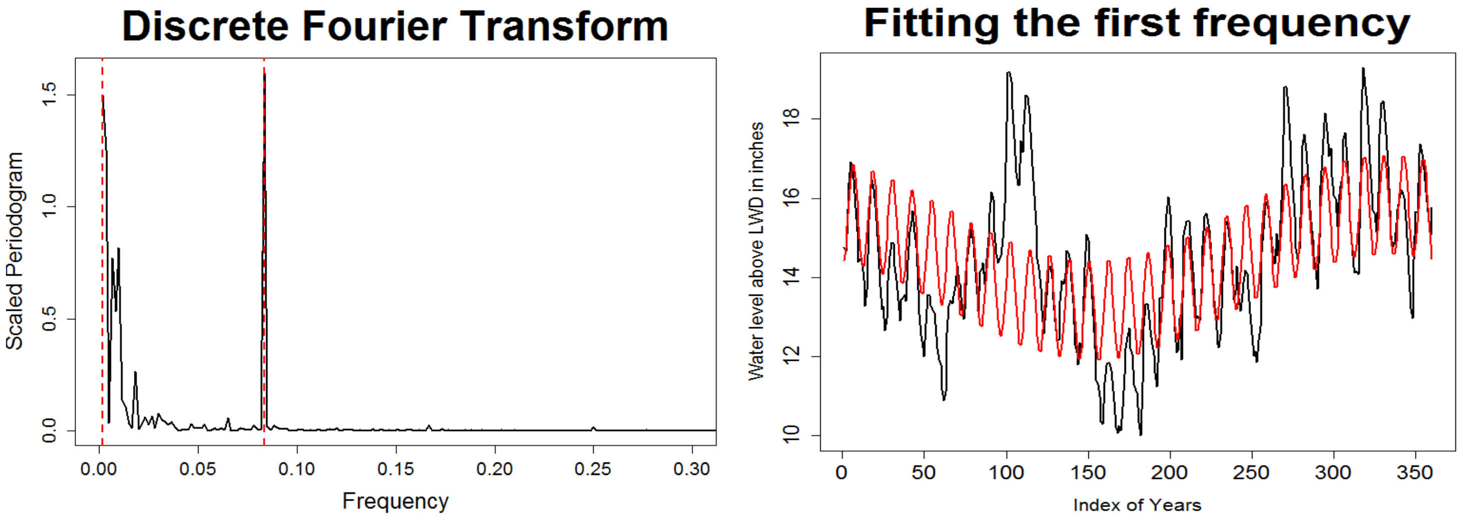


Figure 2: Acf and Pacf of the first 60% of our series

From these two plots we can see that the seasonal component seems to have a period of 12 months and by looking at the entire series, there seems to be a sinusoidal wave driving the general trend of the water level with a period of approximately 30 years. This reflects the non-stationarity of our series. Furthermore, we performed the Augmented Dickey-Fuller test with lag 6, which failed to reject the null hypothesis at 0.05 level that the series has a unit root, which means that our series is non-stationary.

Detrending the Series

Building on our previous discussion, we chose to apply Discrete Fourier Transform to our series.

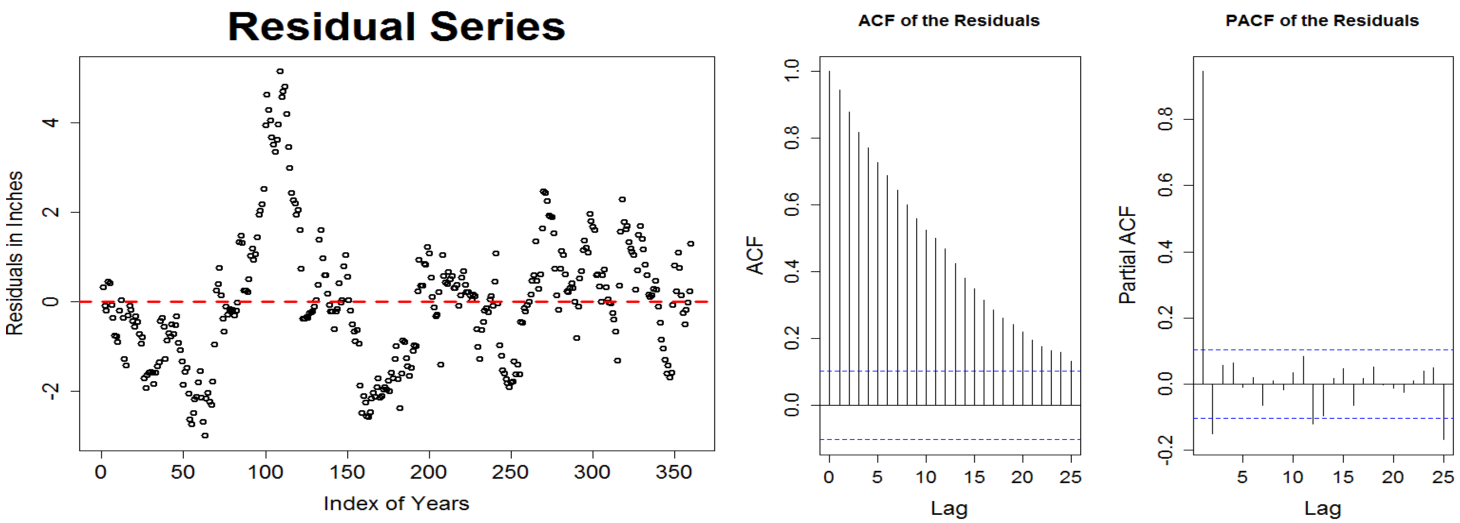


(a) Periodogram with the two dominant frequencies

(b) Detrending the series using the first two dominant frequencies

Figure 3

From the plot of periodogram, we can see that the first two dominant frequencies are 0.00277 and 0.0833 which correspond to periods of 30 years and 12 months, respectively. And by plotting the Fourier series generated by these two frequencies, we can find out that the FFT managed to capture the main trend of the series. Also, with respect to the proportion of variance of the series explained by the Fourier transform, the formula, $\frac{1}{T}\sum_{i=1}^{T}(y_i - \bar{y})^2 = \frac{1}{2}\sum_{j=1}^{M}(\hat{\alpha}_j^2 + \hat{\beta}_j^2)$, where $2M+1 = T$, showed that approximately 50% of the variance of the series is explained by these two frequencies. We know that by including more frequencies in the generation of the Fourier transform, we are able to recover the whole series. To avoid overfitting, we decided to search different combinations of only the first four dominant frequencies. We eventually concluded during our parameter tuning loop that using the combination of the first two frequencies performs the best in terms of one-step ahead rolling forecast.
In order to check whether detrending was successful or not, we created the residual plot and its corresponding Acf and Pacf plots:



(a) Residual Plot

(b) Acf and Pacf plots of the residuals

Figure 4

The residuals appear homoscedastic if you ignore the outlying water levels just before the drought in the 1930's. And since we no longer see evidence of seasonality, we would exclude SARIMA models in our model parameter tuning procedure. However, because the Augmented Dickey-Fuller Test did not reject the null hypothesis of non-stationarity where the p-value equaled 0.09 with lag 6 which was chosen by round(ln(length(the residual series))), we chose to optimize ARIMA models to fit the residual series.

Model Selection and Parameter Tuning

We can demonstrate our model building procedure similar to a water cycle, where we looped through 4 different frequencies for detrending residuals and 18 different ARIMA models from white noise to ARIMA (2,1,2). We decided to stopped at ARIMA(2,1,2) for the sake of parsimony and because the Pacfs of the residual series are not significant after lag 2. We evaluated each one of these 72 unique models by tabulating the mean square error of the one-step ahead rolling forecast for the second section of the original water series.

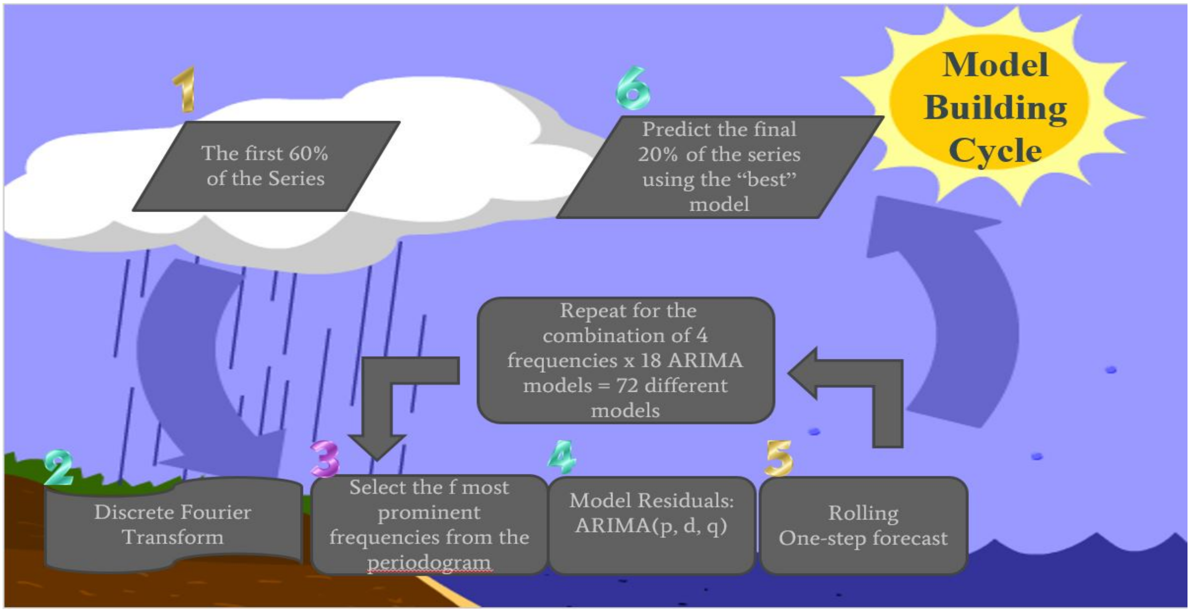

Model Building Cycle
1
The first 60% of the Series
2
Discrete Fourier Transform
3
Select the f most prominent frequencies from the periodogram
4
Model Residuals: ARIMA(p, d, q)
5
Rolling One-step forecast
Repeat for the combination of 4 frequencies x 18 ARIMA models = 72 different models
6
Predict the final 20% of the series using the "best" model

Figure 5: Model Tuning Cycle

By selecting the model with the lowest MSE, our optimal model was ARIMA(2,1,1) using the combination of the first two frequencies to detrend the series. The following plot shows all the MSEs of these 72 models where our "best" model is represented by the dot circled in pink. As we can see from the plot, other than white noise models and pure MA models which resulted in high MSEs, the difference between the remaining models is quite subtle.

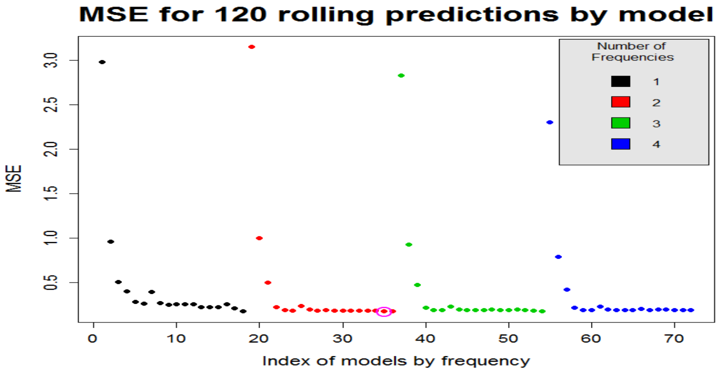


Figure 6: Mean Square Error of 72 Models

Model Assessment

The goal of this exercise was to identify a model which would perform well in one-step ahead rolling forecast of the final 20% hold-out set.

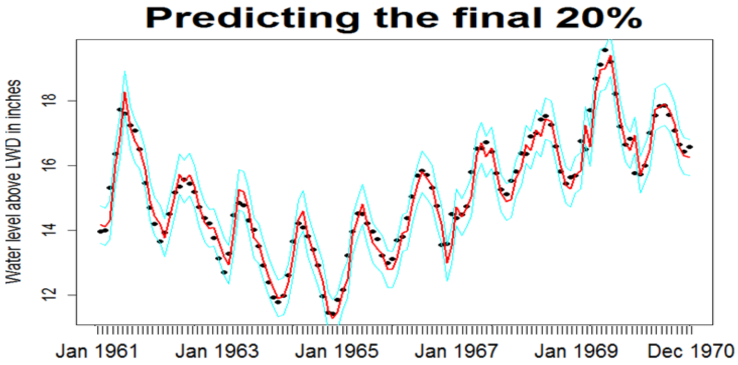


Figure 7: Best Model One-Step Ahead Rolling Forecast

The model we identified during our parameter tuning and model building loop resulted in an MSE of 0.144 sq. inches for rolling one step ahead predictions of the hold-out set. Approximately 89% of the true values were contained within the 80% prediction interval. Additionally, the rolling forecasts differed by no more than 1.2 inches either above or below the true values. We believe that this model is not only quite accurate in predicting the hold-out set, but for practical uses, the model should serve well.

References

[1] Jeffrey A. Ryan, Joshua M. Ulrich. *Package 'xts'*. A library providing uniform handling of R's different time-based data classes. 2013. https://cran.r-project.org/web/packages/xts/xts.pdf.

[2] G. Grothendieck. *Package 'gsubfn'*. Tools for formatting timestamps. 2014. https://cran.r-project.org/web/packages/gsubfn/gsubfn.pdf

[3] G. Grothendieck. *Package 'quantmod'*. Specify, build, trade, and analyse quantitative financial trading strategies. 2015. https://cran.r-project.org/web/packages/quantmod/quantmod.pdf

[4] Michael Weylandt. *Package 'xtsPlots'* Advanced plotting functionality for xts time series objects. 2016. https://github.com/michaelweylandt/xtsPlots

[5] Rob J Hyndman et al. *Package 'forecast'* Methods and tools for displaying and analysing univariate time series forecasts 2015. https://cran.r-project.org/web/packages/forecast/forecast.pdf

[6] Brian Ripley et al. *Package 'MASS'* Functions and datasets to support Venables and Ripley, "Modern Applied Statistics with S" (4th edition, 2002). 2015. https://cran.r-project.org/web/packages/MASS/MASS.pdf

[7] Joshua Ulrich. *Package 'TTR'* Functions and data to construct technical trading rules with R. 2015. https://cran.r-project.org/web/packages/TTR/TTR.pdf

[8] Adrian Trapletti et al. *Package 'tseries'* Time series analysis and computational finance. 2015. https://cran.r-project.org/web/packages/tseries/tseries.pdf

[9] Rmetrics Core Team et al. *Package 'fBasics'* Environment for teaching "Financial Engineering and Computational Finance". 2014. https://cran.r-project.org/web/packages/fBasics/fBasics.pdf

[10] Achim Zeileis, Gabor Grothendieck, Jeffrey A. Ryan, Felix Andrews. *Package 'zoo'* An S3 class with methods for totally ordered indexed observations. 2015. https://cran.r-project.org/web/packages/zoo/zoo.pdf

[11] Wikipedia. *Lake Erie* https://en.wikipedia.org/wiki/Lake_Erie

[12] Professor Katherine B. Ensor. *Lecture Slide 10.*

[13] Robert H. Shumway, David S. Stoffer. *Time Series Analysis and Its Application With R Examples, Third Edition, Blue Printing*, Springer. 2015.

[14] Ruey S. Tsay. *Analysis of Financial Time Series, Third Edition*, Willy . 2010.

[15] James D. Hamilton. *Time Series Analysis*, Princeton University Press. 1994.

Appendix

```r
#########################################################################
## The final 20% prediction loop using the best model
###The output would be one prediction vector and one proportion vector

library(xts); library(gsubfn); library(quantmod)
library(xtsPlots); library(forecast)
library(MASS); library(TTR); library(tseries)
library(fBasics); library(zoo)

## reading in the data and formatting
data<-read.csv('monthly-lake-erie-levels-1921-19.csv',header=TRUE,stringsAsFactors=FALSE)
data<-na.omit(data)
data[,1]<-as.yearmon(data[,1])
data<-xts(data[,-1], order.by=data[,1])

## splitting the data into parts
first <- data[1:360,]; firstX<-as.ts(first)
second <- data[361:480,]; secondX<-as.ts(second)
third <- data[481:600,]; thirdX<-as.ts(third)

### initially vectors for big loop
Freq <- numberOfFreq <- j <- 2; PP <- 2; D <- 1; Q <- 1;
finalPredTable <- as.data.frame(matrix(nrow=length(thirdX), ncol = 1))
finalPropor <- as.data.frame(matrix(nrow = length(thirdX), ncol = 1))
colnames(finalPropor) <- c("F2")
colnames(finalPredTable) <- paste("a(", PP, D, Q,")", "F =", Freq)
ub<-lb<-numeric(length(thirdX))

## start Asssessment loop
for(i in 1:length(thirdX))
```

```
{
  if(i==1)# get the original 80% of the observations
  {
    series<-c(firstX, secondX)
  }else{# add new observations from the hold-out set
    series<-c(firstX, secondX, thirdX[1:(i-1)])
  }
  Ts<-length(series)
  I<-abs(fft(series))^2/Ts
  if(Ts%%2==0)#if n is even
  {
    f<-1:(Ts/2-1)/Ts
    P<-(4/Ts)*I[2:(Ts/2)]
  }else{#if n is odd
    f<-1:{(Ts-1)/2}/Ts
    P<-(4/Ts)*I[2:{(Ts+1)/2}]
  }
  varseries<-var(series)*(1-1/Ts)
  freq <- f[order(P, decreasing = TRUE)[1:j]]
  regMatrix <- matrix(ncol = length(freq)*2, nrow = Ts+1)
  for (l in 1:j){
    regMatrix[,{2*l}-1] <-cos(2*pi* {1:(Ts+1)}*freq[l])
    regMatrix[,2*l] <-sin(2*pi * {1:(Ts+1)}*freq[l])
  }
  reg <- lm(series~regMatrix[-(Ts+1),])
  specPred<-sum(reg$coefficients*c(1,regMatrix[Ts+1,]))
  finalPropor[i,] <-0.5*sum(reg$coefficients[-1]^2)/varseries
  ############## FIT ARIMA Model ################
  tempModel <- arima(x = reg$residuals, c(PP,D,Q))
  finalPredTable[i,] <- specPred+forecast.Arima(tempModel,h=1,level=c(80))$mean
  ub[i]<-specPred+forecast.Arima(tempModel,h=1,level=c(80))$upper
  lb[i]<-specPred+forecast.Arima(tempModel,h=1,level=c(80))$lower
}

######### Calculating MSE ###############
finalMSE<-numeric(dim(finalPredTable)[2])
names(finalMSE)<-colnames(finalPredTable)
finalMSE <- mean((finalPredTable-as.numeric(thirdX))^2)

###############################################################################
## final prediction plot
plot(coredata(thirdX),pch=21,bg="black", main = "Predicting the final 20%",
     cex.main=3, cex.axis=1.3, cex.lab = 1.5, ylab = "Water level
     above LWD in inches")
lines(finalPredTable,col="red",lwd=2)
lines(ub,col="cyan")
lines(lb,col="cyan")
index(third)[which.max(as.matrix(finalPredTable-as.numeric(thirdX)))]
```