Central Coast Ocean Time Series Analysis

Kaden Nichols

Executive Summary

In undergoing this case study of the annual water temperatures of the Santa Barbara Channel from 2020-2022, I hoped to create a forecasting model capable of forecasting future water temperatures and see if the water temperatures had seasonality or cyclical nature to it. Understanding the trends and patterns of the water was the ultimate goal of this analysis. How does water temperature change from year to year? From season to season, does water temperature rotate in some seasonal or cyclical pattern? What else can we learn from a function of local water temperature over the course of the last few years? I began with 3 years of data from 2020-2022, and in order to reduce any initial noise caused by small fluctuations in the data on the ten minute intervals it was originally collected in (and due to computer processing capabilities), had to aggregate the data into daily and then weekly readings. After aggregating the data, I began my analysis of the data and saw immediately that there was a large degree of autocorrelation in the ACF and PACF plots of the time series. After taking the difference of the time series at the first lag, it became easier to interpret the data, and I noted that there may be an autoregressive component of higher order and may not be any moving average component.

This resulted in me comparing a few different models, initially of the higher order autoregressive type AR(23) and some lower order AR() models to see if there was any overfitting.

The questions posed in the context of the study were to understand the patterns our ocean along the South Coast exhibits.

Introduction

Lifestyles on the West Coast have a tendency to mimic the weather, and there is no greater contributor to such weather phenomenon on the West Coast (or the world for that matter), than the Pacific Ocean. Along the California Central Coast, the North Pacific Ocean dictates the humidity, precipitation, wind patterns, and more.

Naturally, I developed an interest in the weather patterns our stretch of coast experiences. Southern California lacks much of the seasonal volatility that the rest of the world experiences. A principal measure of such volatility California does experience is the change in ocean temperature from season to season.

As such, I began to wonder, can the ocean be predicted? Can I take the most recent few years of data and use them to project a forecast of what the next year's water temperature is going to look like? Does the ocean reflect statistically significant seasonality and fluctuations in accordance to the time of year?

Hence I began collecting data from local NOAA NDBC Buoy LLNR196, located 12 Nautical Miles Southwest of Santa Barbara, CA. The data collected is of the years 2020, 2021, and 2022.

Data Analysis and Experimental Design

General Data Information

NOAA is the National Oceanic and Atmospheric Administration, and the NDBC is the group's National Data Buoy Center. Data was specifically taken from NDBC Buoy LLNR196 due to it's proximity to the South Central Coast. The buoy's location is ideal for this analysis as it is both close to the shore and in proximity to Isla Vista, making it's readings pretty reflective of what actually is going on at the beach at my Alma Mater, UCSB. Beyond this, LLNR126 is within the Santa BarbaraChannel, so the presence of the Channel Islands to the South make for the readings to be very specific to the South Coast.

The LLNR196 buoy collected data for the years 2020, 2021, and 2022 was collected every 10 minutes for the entire year. In total, this made for about 150,000 individual observations across the 3 years, taking up 10.2MB of data. Each sample taken includes a reading of the Date (down to the minute), Wind Direction (degTrueNorth), Wind Speed (m/s), Wind Gust (m/s), Wave Height (m), Dominant Wave Period (s), Average Wave Period (s), Mean Wave Direction (degTrueNorth), Air Pressure (hPa at Sea Level), Air Temperature (C), Water Temperature (C), Dewpoint Temperature (C), Visibility (NM), Pressure Tendency (hPa), and Tide (ft).

After converting the data into .csv types, I had to go through and manually convert all data into the desired types, since they had all been character strings when read into R Studio. For the sake of our analysis, I went ahead and removed the columns that were not Water Temperature, but more on that later.

The buoy uses a system of gyroscopes and individual instruments that had been standardized across all of NOAA's buoys in 1998. The NOAA network of ocean buoys comprises one of the most trusted data collection sites for use in a plethora of global sectors both private and public, governmental and civilian. For example, NOAA readings are used for weather forecasting by national weather and news networks, NASA, and the United States military, to name a few.

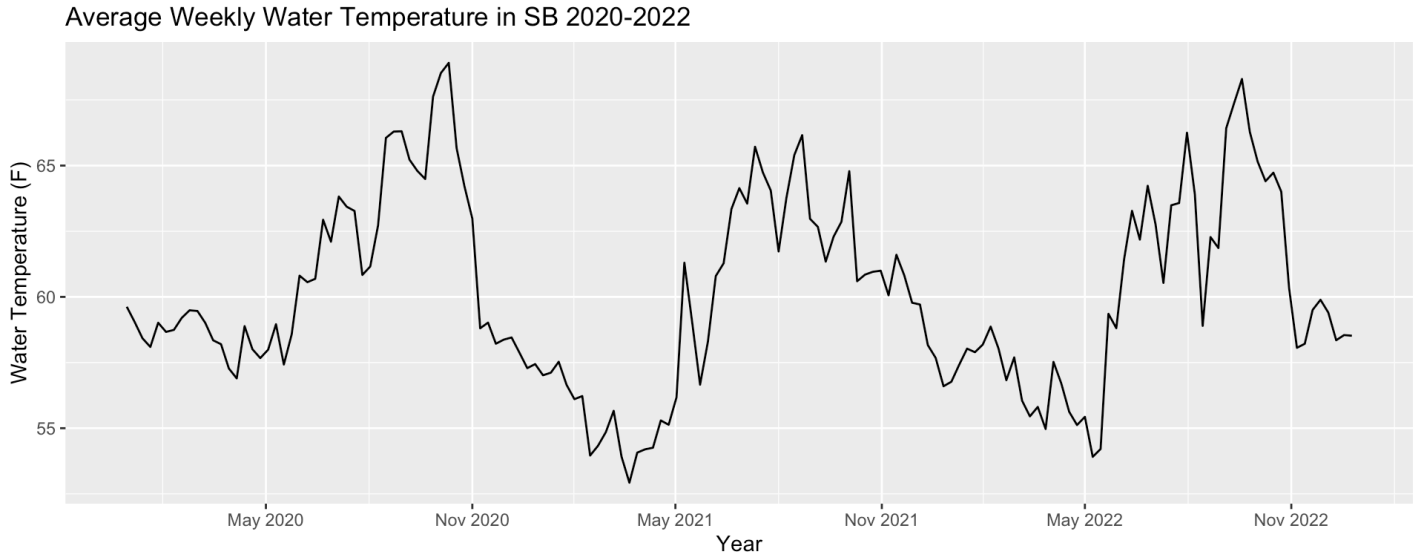
Data Pre-Processing

The initial data sets were separated by year, so the first objective was to chunk them all together and order them by date. For my analysis, I was most interested in just the Water Temperature readings, which are absolutely going to have some correlation with the other variables, (ie. Air Temperature, Air Pressure, etc.) but for the sake of this analysis, I am only considering the water temperature. Also, the water temperature was in units of Celsius, so I had to go through and convert the units to Fahrenheit.

Before appending the data sets together, I had to go through and remove the columns not relevant to the analysis and also combine the columns for Year, Month, Day, Hour, and Minute into just one Date column. At this point, each year's data set had upwards of 50k observations- so I decided to shorten the data set by taking the daily average water temperature reading to reduce each year to 365 observations (366 in the case of 2020 which was a leap year).

Not only did some initial visualizations of my data prove to be computationally taxing, but they also demonstrated an array of small fluctuations that I anticipated becoming a challenge further down the road. As such, I decided to make my data the weekly average of water temperature, instead of the daily averages which stretched my computer too thin. Also, there is a non-technological incentive to doing the analysis of weekly averages over the course of the 3 years, since the average person probably would be unable to distinguish slight changes in water temperature from one day to the next.

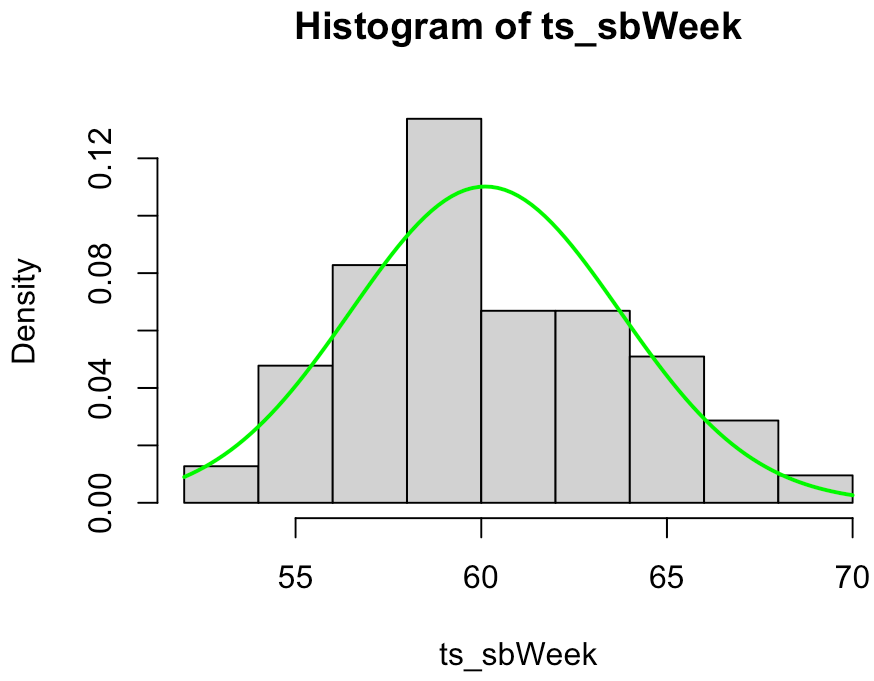
Here is the graph of weekly averages of water temperature over the course of 2020-2022.



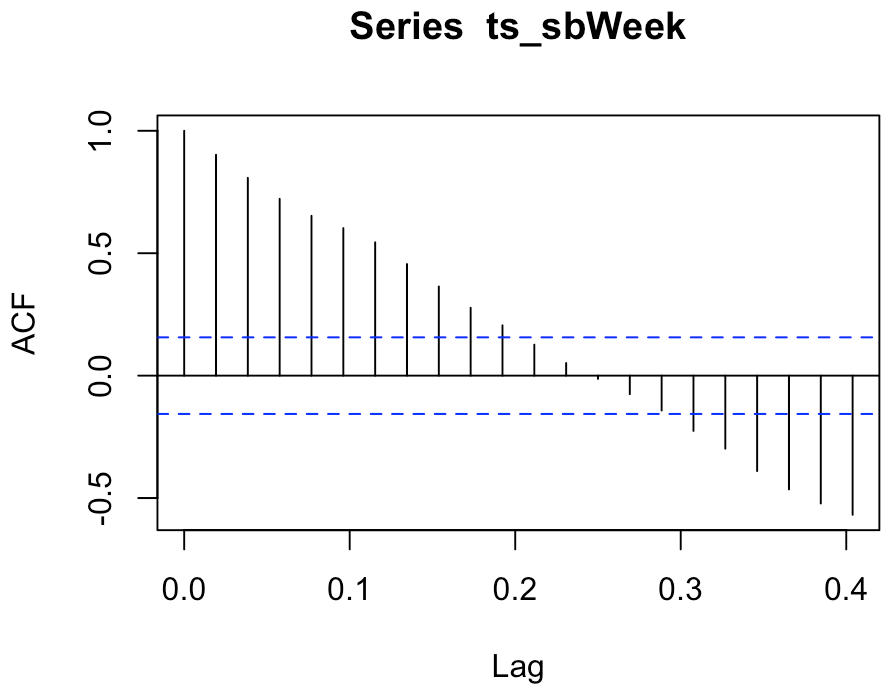
As we can see, there does appear to be some seasonality in accordance to the weather of California, with the end of winter until middle of spring for each year exhibiting some drops in temperature (most likely attributable to natural upwelling) before increasing drastically going into summer. Naturally, we see some semblance of seasonality, but at this point there was no way to know if it was statistically as such.

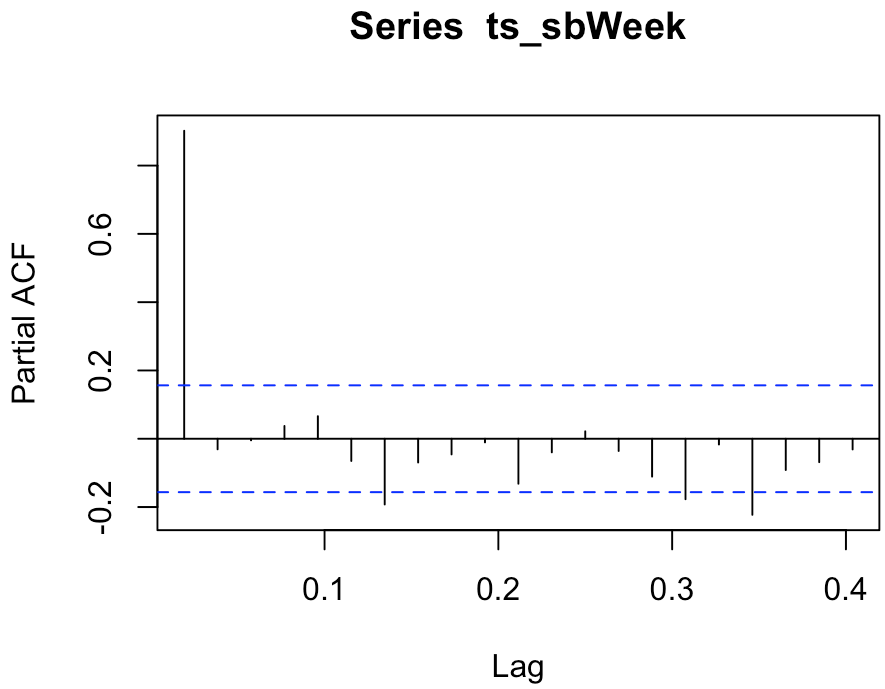
Checking For Normality, ACF, PACF Plots

Before considering the ACF and PACF plots of our time series to determine if there was significant autocorrelative structures or seasonality present, I looked at the histogram of our water temperature readings over the course of the 3 years.



Evidently, our data is somewhat normally distributed, but I expect to be able to improve it much more. Hence I went and looked at the ACF and PACF of our time series to get a better idea of what exactly may be needed to improve our data.





Our ACF plot shows we could potentially do with some transformations of our data, and I assume we could do some differencing to reduce what is evidently some strong autocorrelative structure.

Differencing Our Data

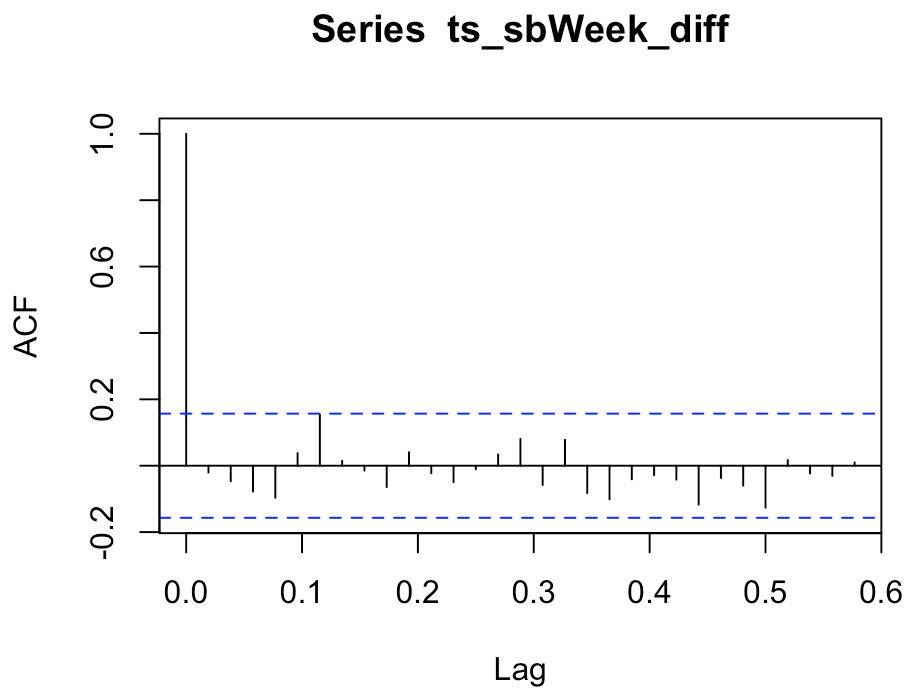
I decided to take the difference of our data at the first lag, and go from there. I looked at the ACF and PACF plots of our new differenced time series and even compared the variances of our initial time series with the new one.

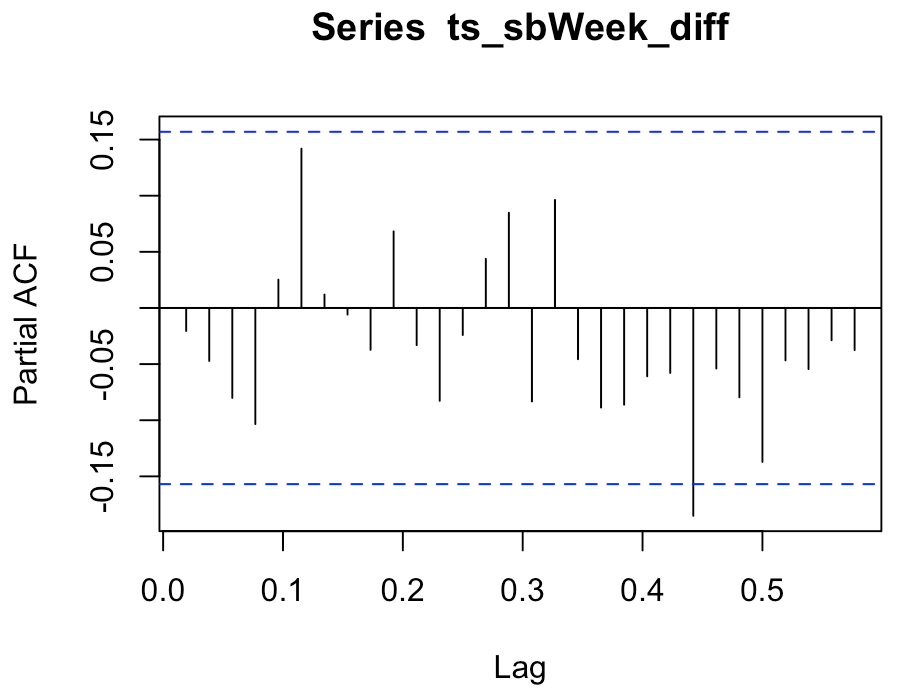
```
var(ts_sbWeek_diff)
```

```
## [1] 2.57338
```

```
var(ts_sbWeek)
```

```
## [1] 13.11438
```



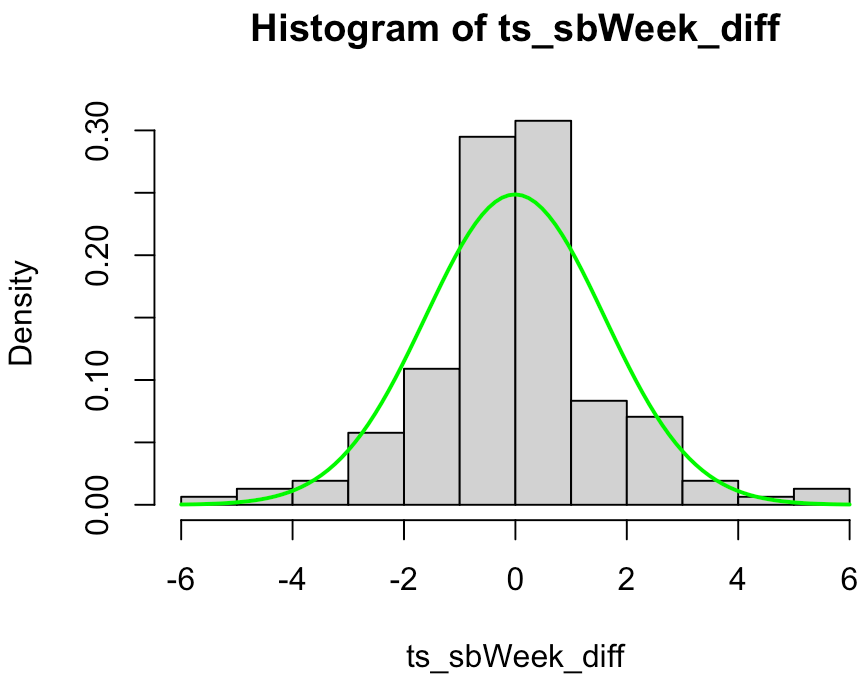


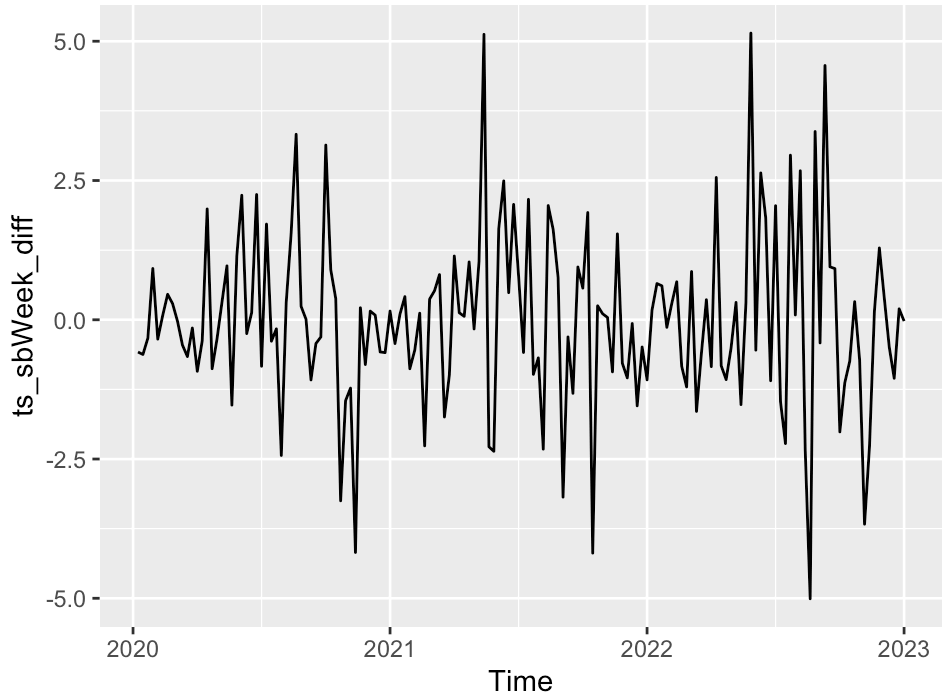
It worked! Taking the difference at lag 1 not only made our ACF show some time series behavior, but also showed that the time series seems to be converging to 0 at higher lags. On top of this, our PACF showed some interesting behavior, with a statistically significant spike at lag 23, which made me start to think that our model may have a MA(0) component and a potentially high order AR(23) component.

As we can see in the ACF and PACF, our differenced data isn't just more appropriate for analysis than our initial time series, but the variance also decreased dramatically, with the initial time series having a variance of 13.11438 and the differenced data having a variance of 2.57338.

```
hist(ts_sbWeek_diff, freq = FALSE)
curve(dnorm(x, mean = mean(ts_sbWeek_diff), sd = sd(ts_sbWeek_diff)),
      add = TRUE, col = "green", lwd = 2)

autoplot(ts_sbWeek_diff)
```



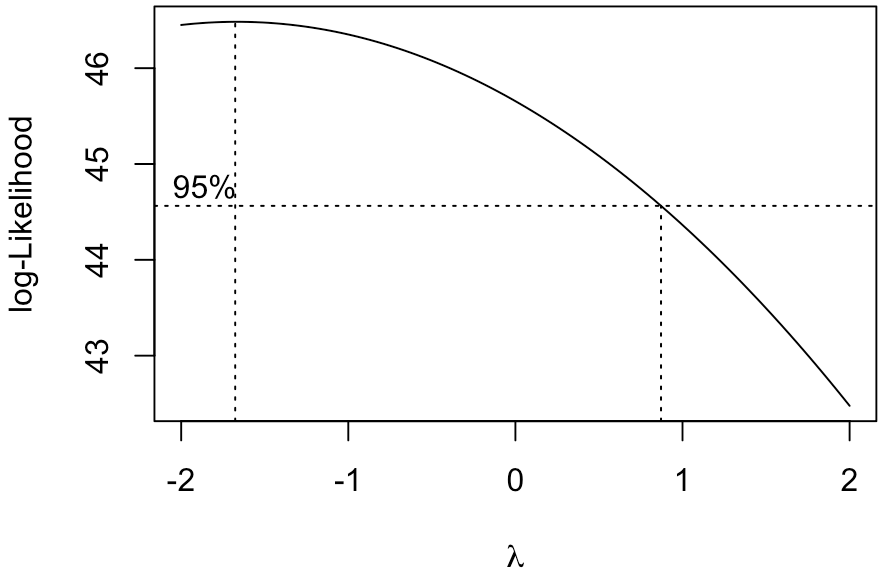


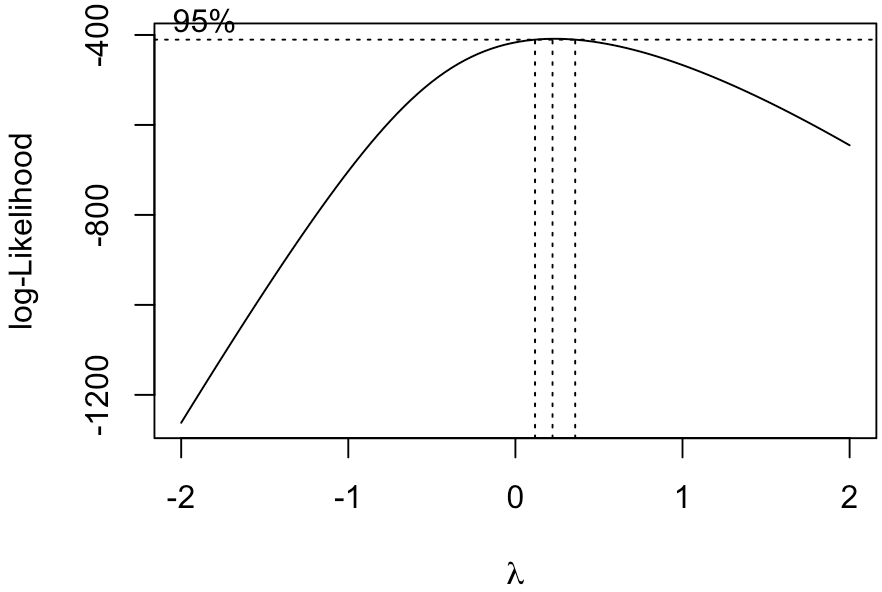
We can see that the differenced data appears more normally distributed, so going forward I am going to be considering it as a potential starting point for building my model.

There is still, however, periods of what seems to be high volatility in our differenced data, so I am going to consider doing some other transformations such as Box Cox, or even taking another difference of our data.

Considering a Box Cox Transformation of our Data

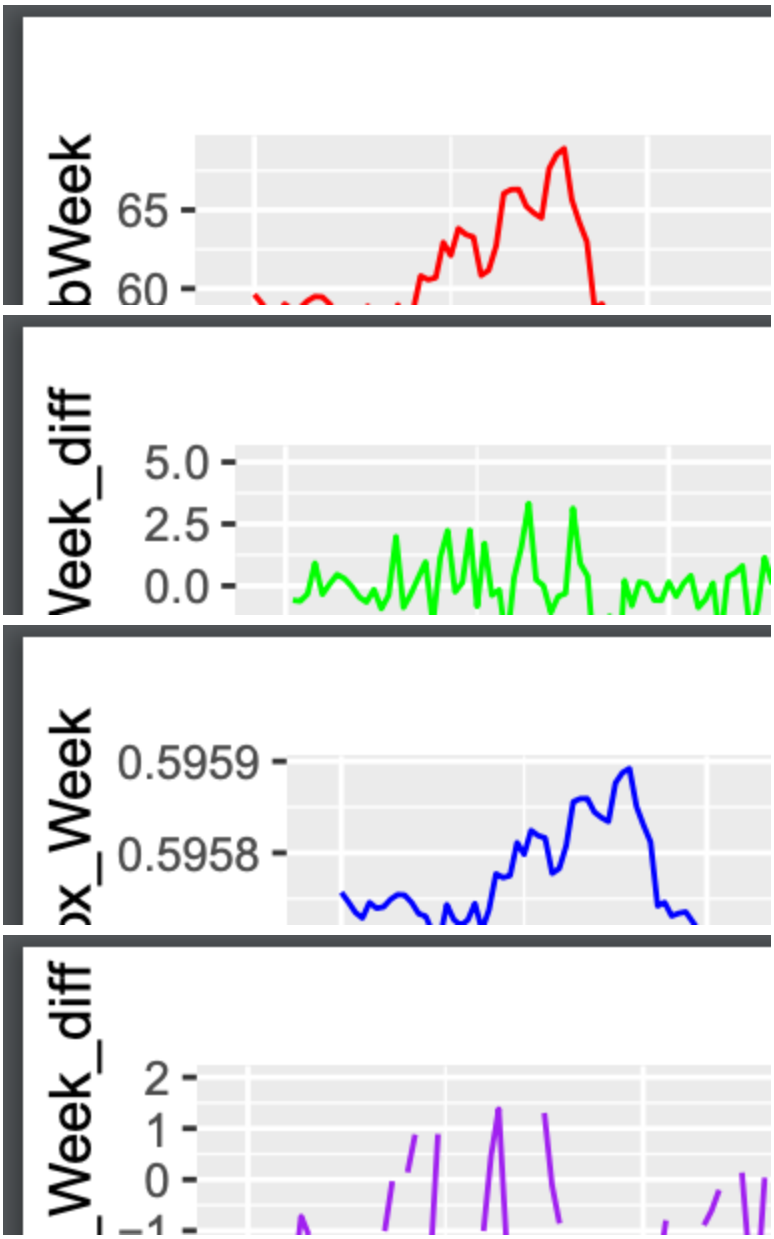
At this point, let's take a look at potentially doing a Box Cox transform of our original data and differenced data to see if we can reduce any excess noise and further decrease our variance.





From our Box Cox plots of first the initial time series, we see that the ideal lambda for a transformation would be about -1.6, but that the confidence interval for it includes 0 and is super wide in the first place.

The Box Cox plot of the differenced data, however, shows a much more narrow confidence interval at a lambda value of about 0.22. I decided to go ahead and transform the data accordingly to see what the effect was and compare how each Box Cox looks in terms of noise reduction.



From the autoplots, we observe that applying the Box-Cox transformation with the optimal lambda values to the original, undifferenced data had minimal effect in smoothing the curve. Additionally, applying the transformation to the differenced time series introduced some discontinuities. This is likely because, as lambda approaches zero, the Box-Cox transformation behaves similarly to the logarithmic function. Since the differenced time series contains negative values, they cannot be transformed in the desired manner

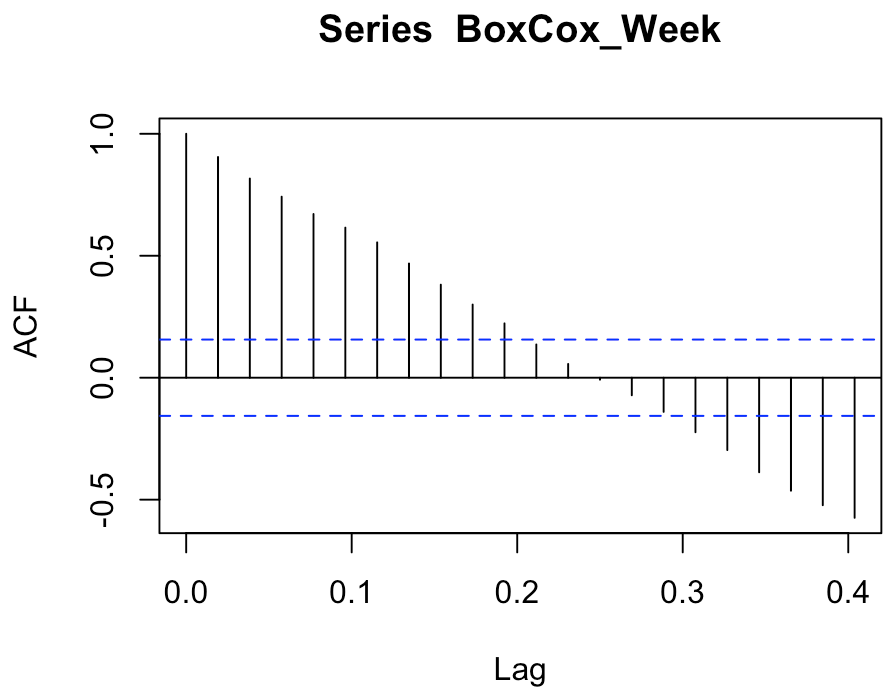
We observe that the differenced time series somewhat resembles white noise, and the Box-Cox transformation of the original data appears to have significantly reduced variance. However, the Box-Cox transformation applied to the differenced data seems unable to fully capture the behavior of the model due to the presence of such discontinuities.

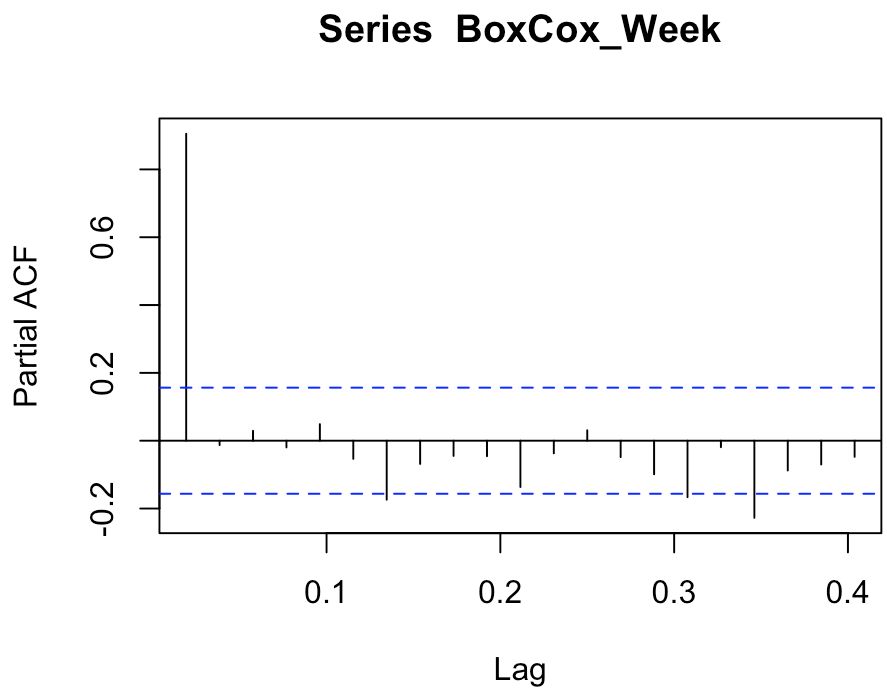
Now, let's consider how the variance of the Box Cox transformation of the original set compares to the others.

```
var(BoxCox_Week)
```

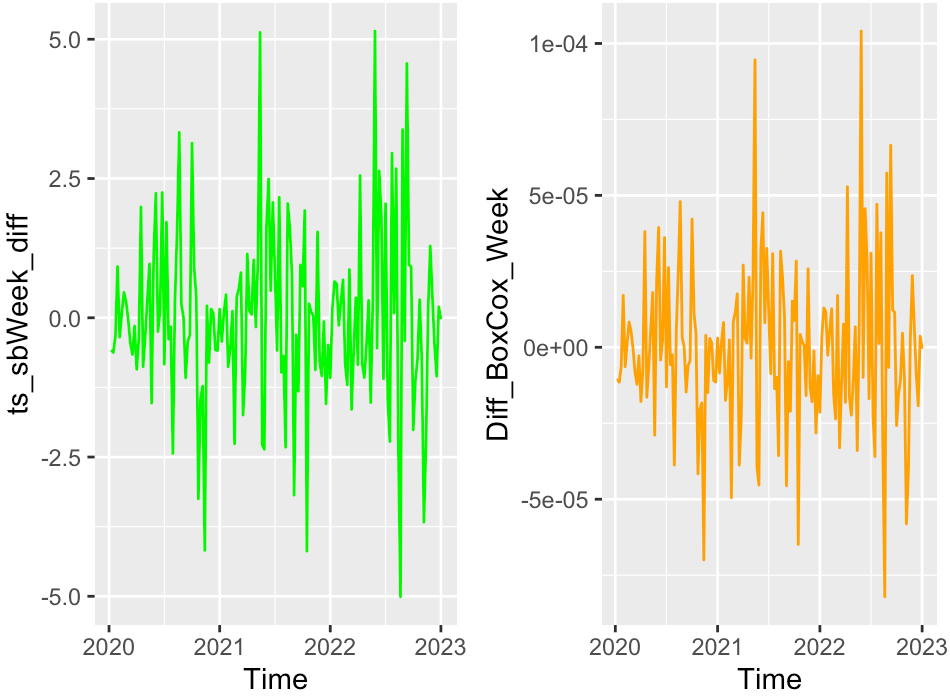
```
## [1] 3.860128e-09
```

The variance of the Box Cox transformation is drastically lower than both the differenced time series and the initial time series, so I am going to move forward and consider using it for a potential model.

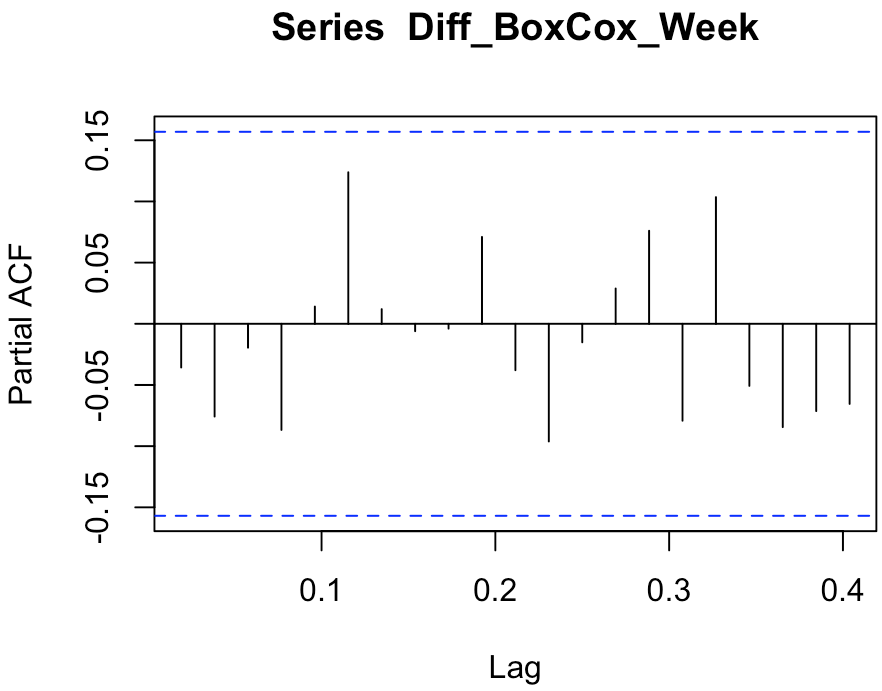


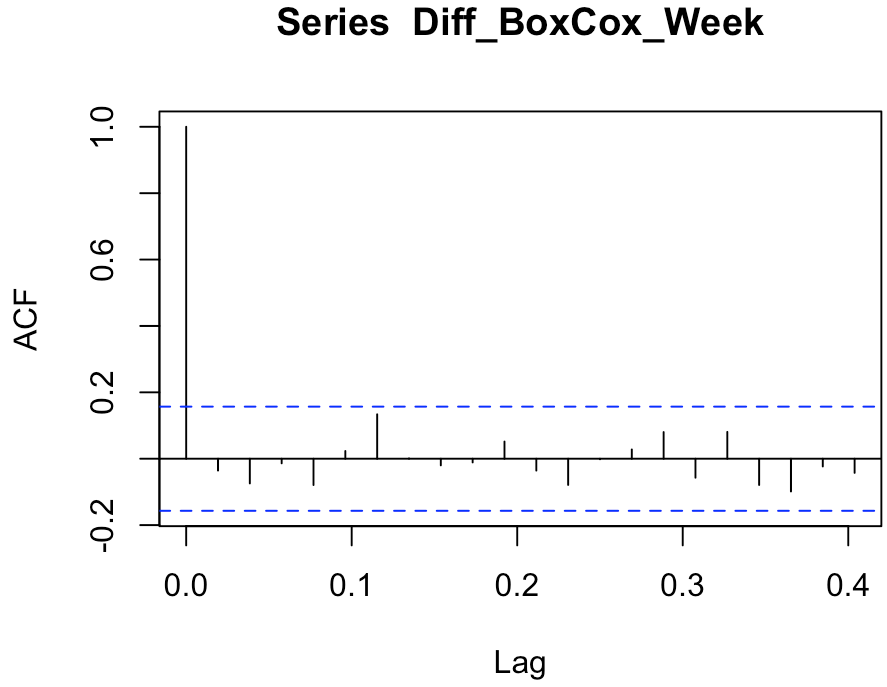


Yikes! The Box Cox transformation is still displaying a whole bunch of autocorrelative structure, so I will proceed and take the difference of it to see if we can reduce the background noise which may cause some issues further down the road if we opt to incorporate it in our model.



The two differenced models appear very similar, but the variance of the differenced Box Cox transform is less than that of the initial difference of the un-transformed time series data. Now to take a look at the ACF and PACF of our Differenced Box Cox transform.





The ACF and PACF for our Differenced Box Cox transform displays the ACF decay to 0 with no significant spikes and the PACF also has no significant spikes at any lag. A concern I have at this point is that I am potentially over-correcting the data by transforming it so much, so moving forward I will test for stationarity and the presence of significant autocorrelation for the Differenced Original Time Series and the Differenced Box Cox we have been working with.

Testing for Stationarity and Presence of Significant Autocorrelation

Augmented Dickey Fuller

The Augmented Dickey Fuller test for stationarity will allow us to determine if the time series we are working with are stationary, which is vital in building a viable forecasting model. If the test concludes that a series is not stationary, then we ought to further refine the model and capture more desirable statistical properties essential to developing a reliable forecasting model such as time independent mean and variance.

Null Hypothesis: Series is non-stationary or series has a unit root. Alternative Hypothesis: Series is stationary and lacks a unit root

We will look to reject the null hypothesis if the p-value of the ADF test is lower than our significance level of 0.05.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_sbWeek_diff
## Dickey-Fuller = -4.6542, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Diff_BoxCox_Week
## Dickey-Fuller = -4.6566, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

The p-val of the ADF test on the differenced time series and the Box Cox differenced series is less than our significance level of 0.05. Hence we move to reject the Null Hypothesis that our series are in fact stationary. Both are still considered viable options.

We will now move on to testing them in the Ljung Box Test which tests for significant autocorrelative structure.

Ljung Box Test

We will conduct a Ljung Box test on our time series to see if they resemble white noise, which would imply a lack of autocorrelative structure in the residuals and be a good sign for proceeding to building a model with the time series we have. To proceed with a model for forecasting, we want our residuals to resemble white noise.

Null Hypothesis: The residuals resemble white noise. (ie. No significant autocorrelation) Alternative Hypothesis: The residuals are not white noise. (ie. Model is not properly capturing the structure of the time series.)

We will look to reject the Null Hypothesis if the p-value of the Ljung-Box test is less than the significance level of 0.05.

```
##
##  Box-Ljung test
##
## data:  ts_sbWeek_diff
## X-squared = 0.06663, df = 1, p-value = 0.7963
```

```
##
##  Box-Ljung test
##
## data:  Diff_BoxCox_Week
## X-squared = 0.20286, df = 1, p-value = 0.6524
```

The p-val of our Ljung Box Test for both possible models are greater than the significance level, so we fail to reject the Null Hypothesis in both cases and conclude that the model's both resemble white noise. As such, we will proceed with our models as viable for building a forecast.

KPSS Test

We will now examine the stationarity assumptions of our models by testing for stationarity in the presence of a trend. If the models are not stationary, any forecasts built from them will likely fail to accurately capture the behavior of the time series, diminishing overall accuracy.

Null Hypothesis: Time series is trend stationary Alternative Hypothesis: Time series is not trend stationary

We will move to reject the Null Hypothesis if the p-value is less than the significance level of 0.05.

```
##
##  KPSS Test for Trend Stationarity
##
## data:  Diff_BoxCox_Week
## KPSS Trend = 0.044069, Truncation lag parameter = 4, p-value = 0.1
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  ts_sbWeek_diff
## KPSS Trend = 0.04316, Truncation lag parameter = 4, p-value = 0.1
```

Both of our models have p-values listed as greater than 0.1, which is greater than 0.05. Thus we can conclude that both models are in fact trend stationary.

Testing Conclusions

For the differenced model of the original time series, we have seen with the results of the Augmented Dickey Fuller test and the KPSS test that it is in fact stationary. Likewise, the results of the Ljung Box test allow us to conclude that the residuals time series are in fact resemblant of white noise and we can expect to be able to proceed and build a forecast with this model.

For the differenced box cox model, on the other hand, we also concluded that the model is stationary through our use of the Augmented Dickey Fuller test and the KPSS test. With the results from the Ljung Box test, we have also concluded that the residuals of this model are white noise and we can expect to be able to create a forecast with this model.

Let's go ahead and start fitting actual ARIMA models accordingly.

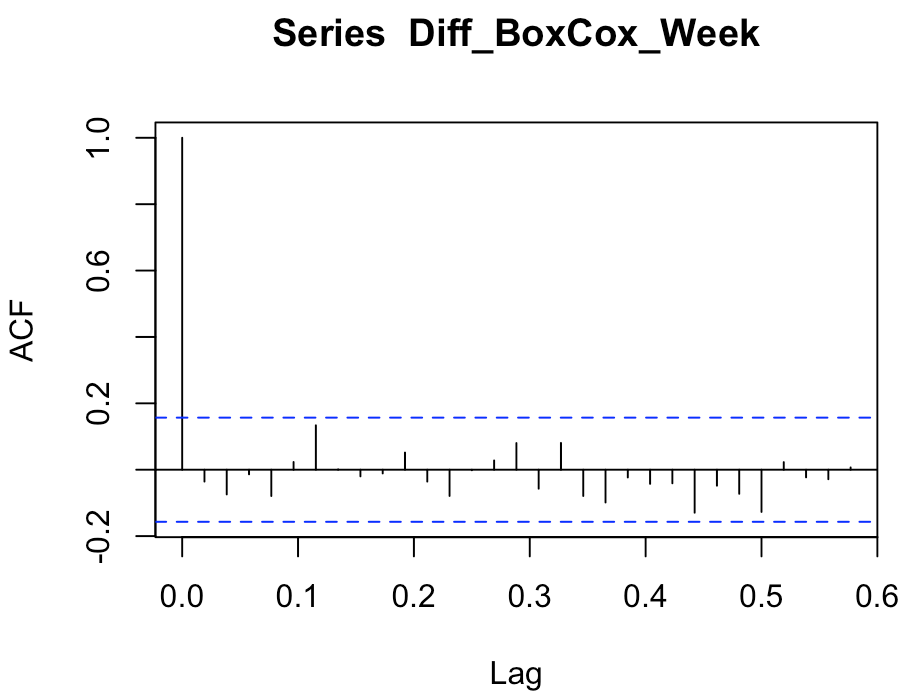
Time Series Modeling and Forecasting

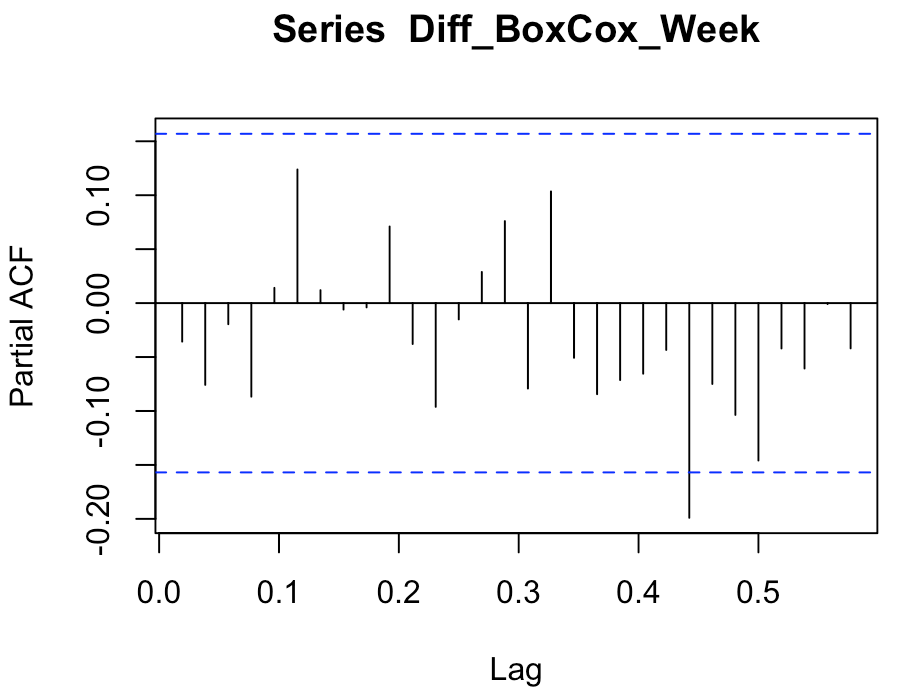
Let's go ahead and take a look at what we have gathered thus far.

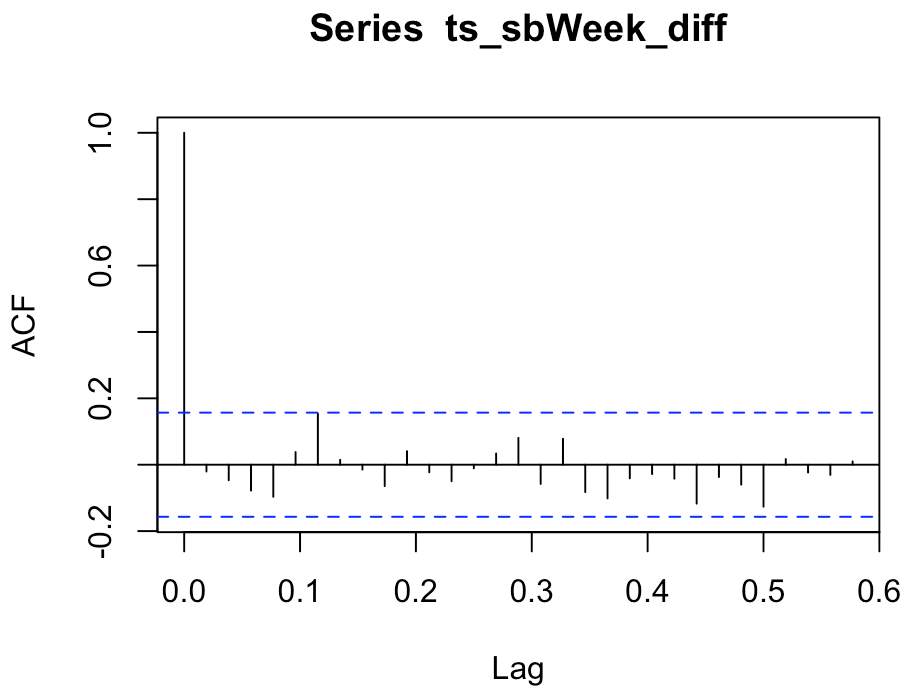
Quick Recap of our Analysis Thus Far

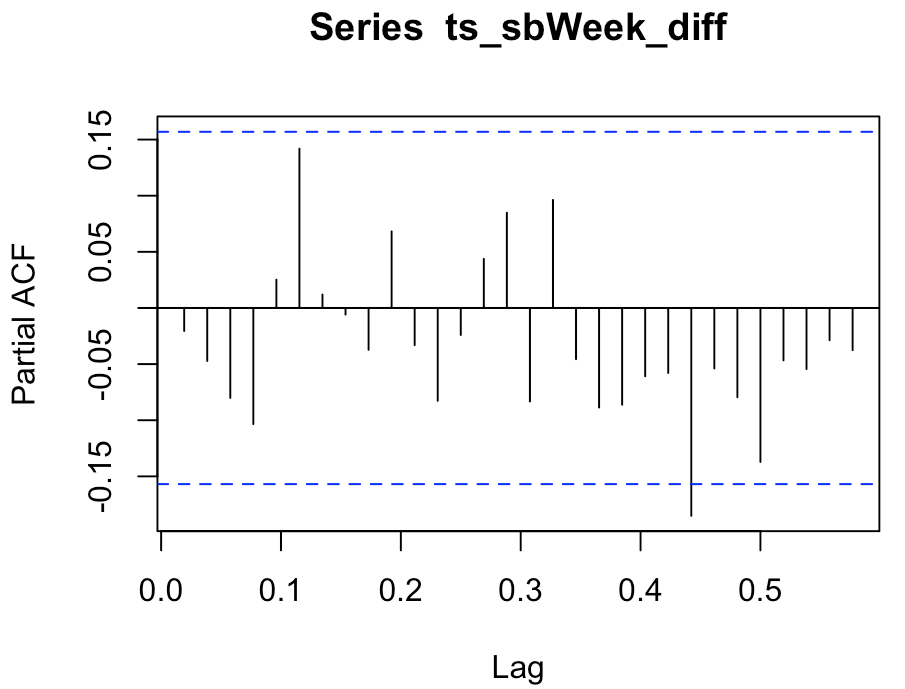
We started with a handful of potential models for our time series, and did some quick empirical analysis of some of the properties of each potential model using the ACF and PACF plots of each, as well as conducted some hypothesis tests to see if the models we decided were most appropriate were statistically reasonable to apply to a forecasting model. We tested for the stationarity of each of our two prospective time series and for the lack of autocorrelation in the residuals of each as well.

Now, we can proceed and actually begin building our models with our time series in order to make some forecasts of future water temperatures in Santa Barbara. Let's look back at the ACF and PACF plots of both our time series- the first of which is the differenced original time series, and the second is the differenced Box Cox transformation.


Series Diff_BoxCox_Week
ACF
1.0
0.6
0.2
-0.2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Lag


Series Diff_BoxCox_Week
Partial ACF
0.10
0.00
-0.10
-0.20
0.0
0.1
0.2
0.3
0.4
0.5
Lag





According to the ACF of both our time series, we do not have any significant spikes visible, but in the PACF of both, we can see that there is a significant spike at lag 23. This goes to show that each of these is most likely gong to have a AR() term but lack a MA() term.

I will first fit an AR(23) model to each time series we are working with.

Fitting the Initial Models

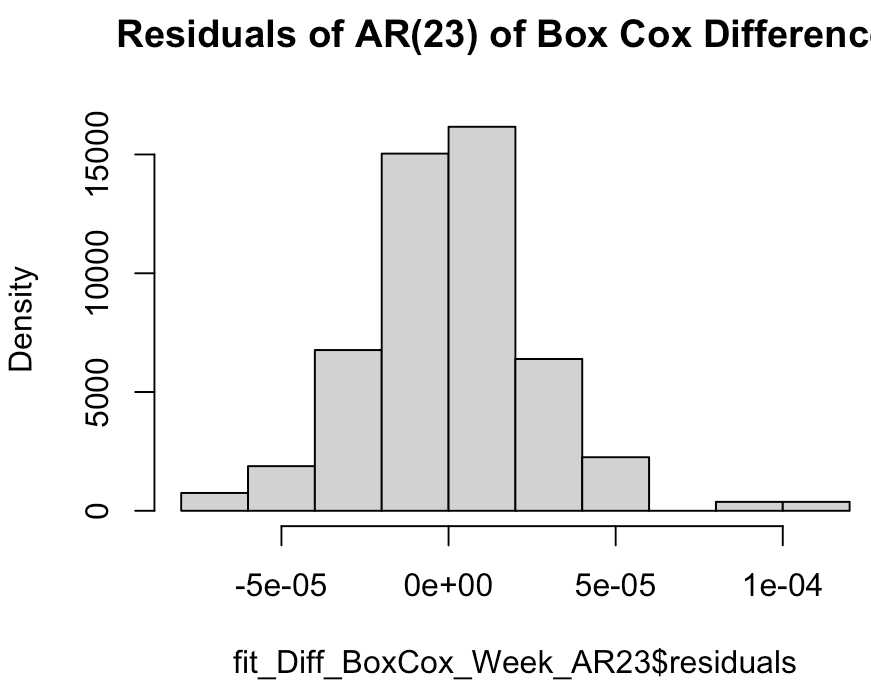
Initial AR()

```
##
## Call:
## tseries::arma(x = ts_sbWeek_diff, order = c(23, 0, 0))
##
## Model:
## ARMA(23,0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.92384 -0.89521 -0.07917  0.81388  5.04927
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ar1       -0.050910    0.078056   -0.652  0.51426
## ar2       -0.043064    0.077443   -0.556  0.57816
## ar3       -0.104320    0.077266   -1.350  0.17697
## ar4       -0.137006    0.076641   -1.788  0.07384 .
## ar5        0.051749    0.076554    0.676  0.49905
## ar6        0.204073    0.077025    2.649  0.00806 **
## ar7       -0.033023    0.078076   -0.423  0.67232
## ar8       -0.006293    0.078288   -0.080  0.93593
## ar9       -0.039067    0.080202   -0.487  0.62619
## ar10       0.071410    0.080025    0.892  0.37221
## ar11      -0.077666    0.080076   -0.970  0.33210
## ar12      -0.092951    0.079652   -1.167  0.24323
## ar13       0.037753    0.079746    0.473  0.63592
## ar14       0.065104    0.079369    0.820  0.41206
## ar15       0.096320    0.080339    1.199  0.23056
## ar16      -0.068027    0.080695   -0.843  0.39922
## ar17       0.150286    0.082886    1.813  0.06981 .
## ar18      -0.062387    0.081585   -0.765  0.44446
## ar19      -0.117967    0.083543   -1.412  0.15793
## ar20      -0.135845    0.086941   -1.563  0.11817
## ar21      -0.090588    0.088786   -1.020  0.30758
## ar22      -0.082270    0.089359   -0.921  0.35723
## ar23      -0.256336    0.089199   -2.874  0.00406 **
## intercept  0.001016    0.124975    0.008  0.99351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 2.414,  Conditional Sum-of-Squares = 318.68,  AIC = 628.2
```

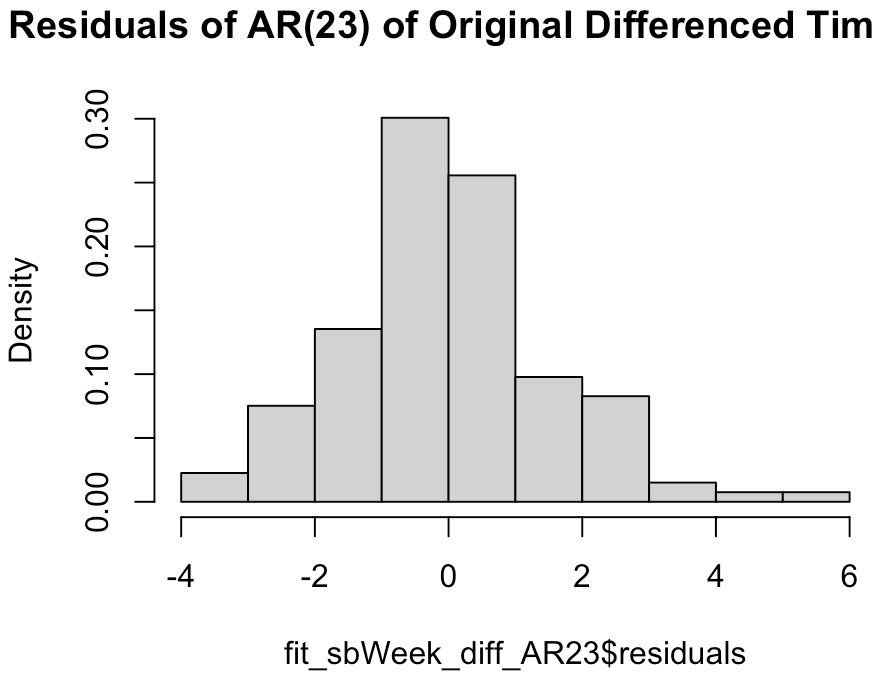
```
##
## Call:
## tseries::arma(x = Diff_BoxCox_Week, order = c(23, 0, 0))
##
## Model:
## ARMA(23,0)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.348e-05 -1.530e-05  1.407e-07  1.309e-05  1.011e-04
##
## Coefficient(s):
##             Estimate  Std. Error  t value Pr(>|t|)
## ar1       -4.993e-02   7.778e-02   -0.642  0.52094
## ar2       -7.268e-02   7.722e-02   -0.941  0.34658
## ar3       -5.241e-02   7.712e-02   -0.680  0.49674
## ar4       -1.184e-01   7.650e-02   -1.548  0.12173
## ar5        4.393e-02   7.629e-02    0.576  0.56472
## ar6        1.841e-01   7.668e-02    2.401  0.01636 *
## ar7       -3.676e-02   7.744e-02   -0.475  0.63497
## ar8       -2.377e-03   7.799e-02   -0.030  0.97569
## ar9        5.601e-03   7.953e-02    0.070  0.94386
## ar10       7.544e-02   7.935e-02    0.951  0.34177
## ar11      -9.004e-02   7.931e-02   -1.135  0.25623
## ar12      -1.107e-01   7.851e-02   -1.410  0.15841
## ar13       5.311e-02   7.863e-02    0.675  0.49942
## ar14       5.040e-02   7.828e-02    0.644  0.51968
## ar15       8.266e-02   7.885e-02    1.048  0.29447
## ar16      -6.319e-02   7.910e-02   -0.799  0.42434
## ar17       1.488e-01   8.129e-02    1.830  0.06723 .
## ar18      -6.265e-02   8.063e-02   -0.777  0.43720
## ar19      -1.156e-01   8.231e-02   -1.405  0.16007
## ar20      -9.166e-02   8.592e-02   -1.067  0.28609
## ar21      -9.172e-02   8.702e-02   -1.054  0.29185
## ar22      -6.262e-02   8.714e-02   -0.719  0.47243
## ar23      -2.661e-01   8.687e-02   -3.063  0.00219 **
## intercept -1.359e-07   2.114e-06   -0.064  0.94875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 6.922e-10,  Conditional Sum-of-Squares = 0,  AIC = -2799.5
```

When we fit the models with an AR(23), a concern I have is that we are overfitting our data. And as we look at the AIC of each fit, we see that the differenced time series model has an AIC of 628.2, whereas the Box Cox difference model has an AIC of -2799.5.

If we were to only consider the AIC values, then the latter model fit would be much more ideal with it's much lower AIC of -2799.5. Before saying anything more about these models, we can check to ensure that the residuals approximate a normal distribution. If they do not, then we will not proceed with such models anyway and reevaluate.



The residuals of this Box Cox AR(23) do in fact represent a normal distribution.Now let's look at the residuals of the other model.



Likewise, these residuals also follow a normal distribution. We do not have to rule either out on the basis of the residuals not being normally distributed.

This disparity between models does raise the concerns about overfitting, so I will go ahead and fit some more AR() models to our series, but of a lower order to compare to the AR(23).

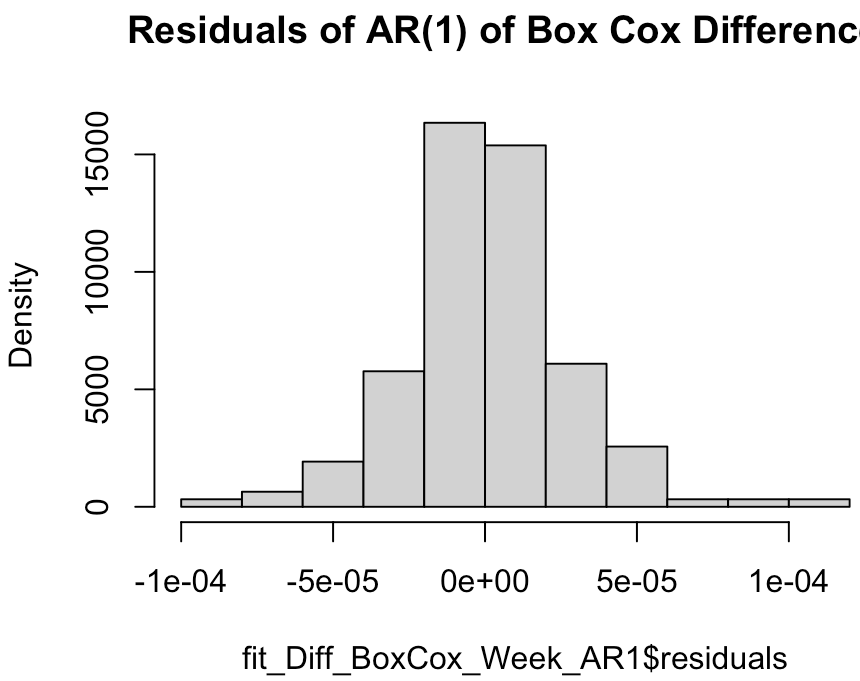
Lower Order AR()

```
## Series: ts_sbWeek_diff
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##           ar1     mean
##       -0.0204  -0.0070
## s.e.   0.0798   0.1255
##
## sigma^2 = 2.589:  log likelihood = -294.55
## AIC=595.09   AICc=595.25   BIC=604.24
##
## Training set error measures:
##                         ME     RMSE      MAE     MPE    MAPE      MASE
## Training set -7.386723e-05 1.598691 1.155038 100.529 100.529 0.6700084
##                      ACF1
## Training set -0.00107456
```

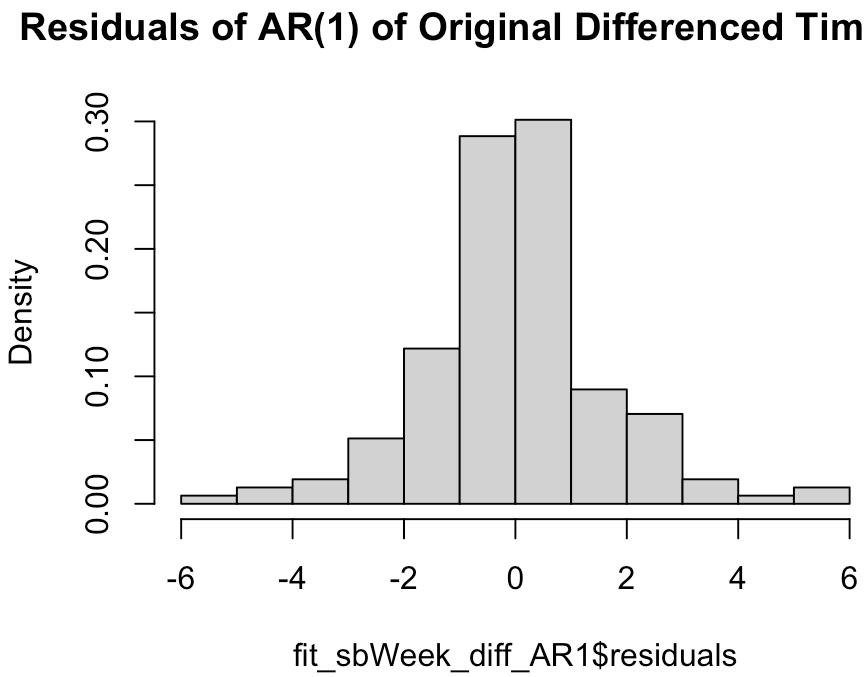
```
## Series: Diff_BoxCox_Week
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##           ar1   mean
##       -0.0355  0e+00
## s.e.   0.0799  1e-04
##
## sigma^2 = 7.387e-10:  log likelihood = 1419.69
## AIC=-2833.38   AICc=-2833.23   BIC=-2824.24
##
## Training set error measures:
##                         ME         RMSE         MAE      MPE     MAPE      MASE
## Training set -2.291584e-09 2.700375e-05 1.96058e-05 100.7235 100.7235 0.6612919
##                       ACF1
## Training set -0.002887957
```

Our new, reduced AR(1) model of the differenced time series has an AIC= 595.09, and the reduced model of the differenced box cox time series has an AIC= -2833.38. Both of these models have improved AIC compared to their respective AR(23) models we just fit earlier, supporting the idea that the AR(23) is of too high an order and may not be the best for modeling the data.

To be sure, I want to make sure that the new AR(1) models are viable and that they also have normally distributed error terms.



The residuals of this Box Cox AR(23) do in fact represent a normal distribution.Now let's look at the residuals of the other model.



From these histograms , we actually can see that not only are the residuals of the models normally distributed (hence white noise), but they are also apparently more normally distributed than the residuals from the AR(23) models. They appear to be normal about the mean with constant variance in both cases.

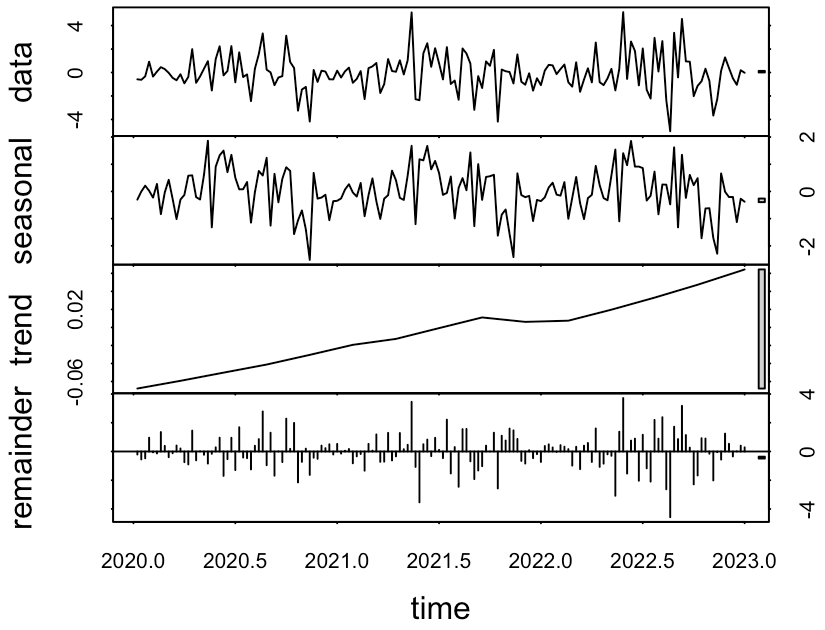
Taking into account the AIC of each of the 4 models we have looked at, I will be proceeding with the models of AR(1) order.

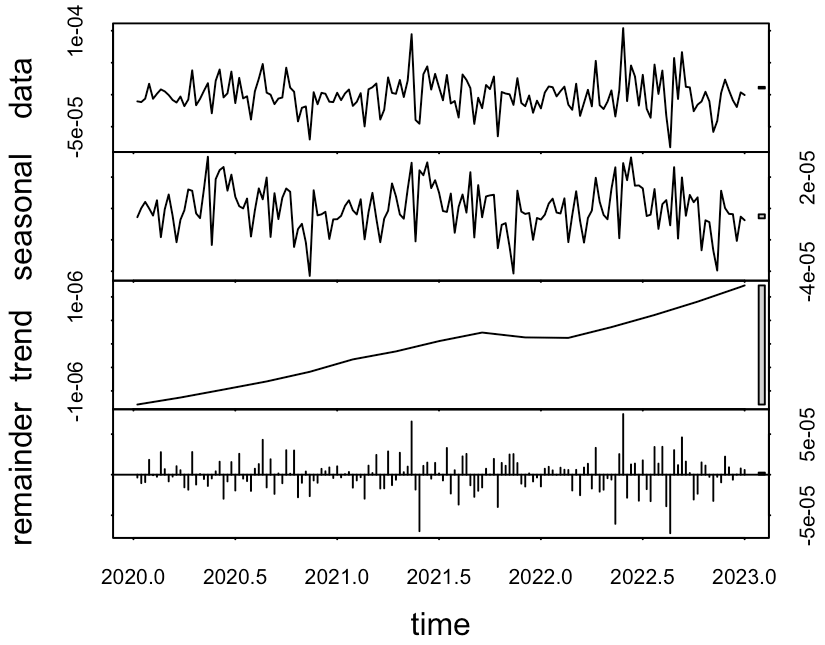
Model Assumptions

The AR(1) models that we will consider are considered autoregressive models of order 1, meaning that the present value of the time series is dependent upon the value of the time series at the time stamp immediately before. In the context of our models, this means that the water temperature is dependent upon the water temperature from a week prior. Our AR(1) models also assume that our residuals are uncorrelated with one another and normally distributed about a mean of 0 with constant variance. This assumption of the distribution of the residuals can be seen in the histograms above, however to be sure we will conduct a Ljung Box test on our residuals to determine if our models have any autocorrelation which we have not yet dealt with.

Visual Of Decomposition

```
plot(stl(ts_sbWeek_diff, s.window=7))
plot(stl(Diff_BoxCox_Week, s.window=7))
```





Interesting, we can note some positive upward trend. Might this be because of some factor such as global warming?

Portmanteau Tests of the Residuals of the AR(1) Model

As from above, we have:

Null Hypothesis: The residuals resemble white noise. (ie. No significant autocorrelation) Alternative Hypothesis: The residuals are not white noise. (ie. Model is not properly capturing the structure of the time series.)

We will look to reject the Null Hypothesis if the p-value of the Ljung-Box test is less than the significance level of 0.05.

First the Ljung Box test of our AR(1) model of the differenced time series.

```
##
##  Box-Ljung test
##
## data:  fit_sbWeek_diff_AR1$residuals
## X-squared = 0.00018362, df = 1, p-value = 0.9892
```

Our p-value is 0.9892, so we can say with a high degree of confidence that the residuals do in fact resemble white noise, which supports our assumption for the AR(1) model. (ie. Fail to reject null hypothesis, conclude residuals are white noise)

Next we can take a look at the Ljung Box test for the differenced Box Cox model.

```
##
##  Box-Ljung test
##
## data:  fit_Diff_BoxCox_Week_AR1$residuals
## X-squared = 0.0013263, df = 1, p-value = 0.9709
```

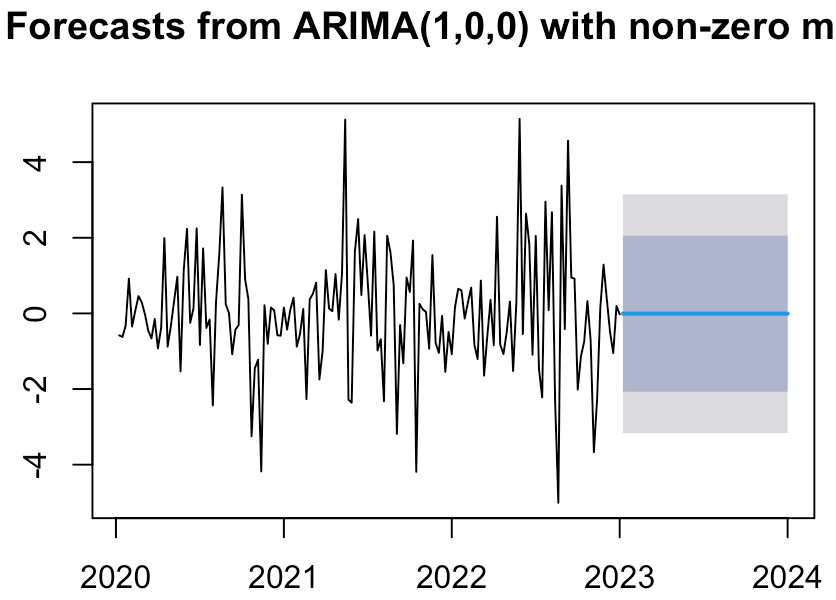
Our p-value for this test is 0.9709, so we can also say with a high degree of certainty that the assumptions for residuals of this AR(1) model are white noise, and that we absolutely fail to reject the null hypothesis.

Understanding Our Models

We utilized the ACF and PACF of our models to help determine what our ARIMA() or ARMA() models would look like. We fit AR(23) models to both of our potential time series and even lower order AR(1) models to them as well. In comparing the AIC of our models, we take note that the AR(1) models for both time series are improved (lower AIC) than their higher order analogous AR(23) models. This maybe attributable to them being overfit at higher autoregressive orders with an AR(23).

Forecasting

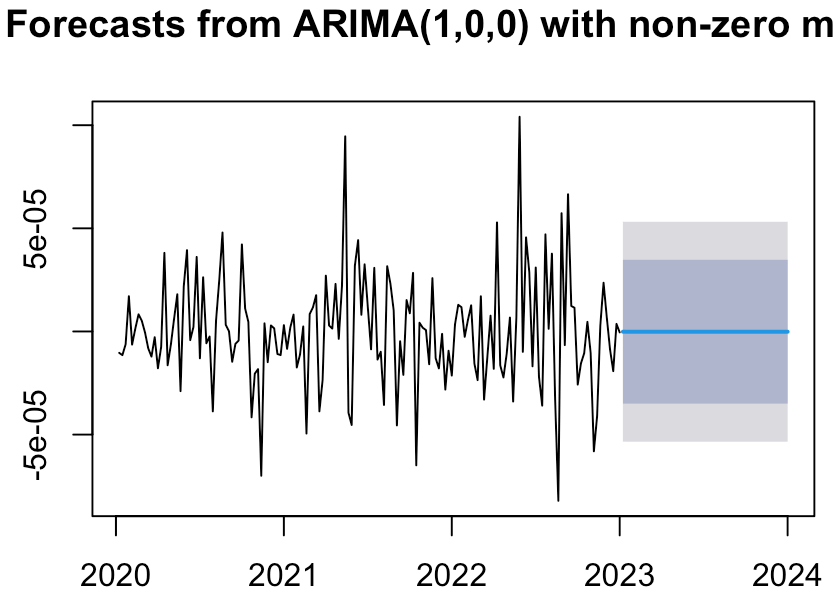
Differenced Time Series



Yikes, this model is actually seemingly very over fit. Perhaps the other differenced Box Cox model is better.

Box Cox Time Series

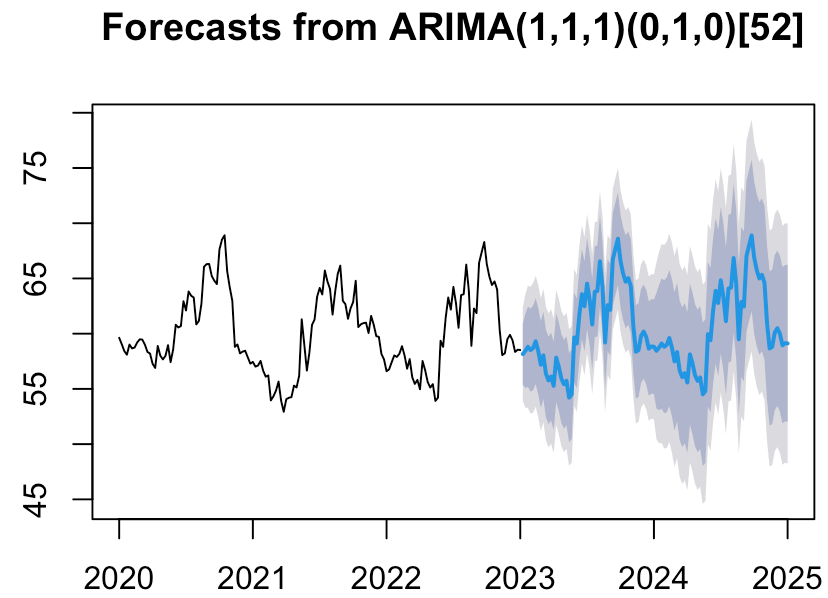
```
#fcast_sbWeek_diff <- forecast(fit_Diff_BoxCox_Week_AR1, h=52)
plot(forecast(fit_Diff_BoxCox_Week_AR1, h=52))
```



It seems that our in attempting to make our time series better suited for analysis and building a forecasting model, we over corrected severely in trying to reduce variance and background noise.

Discussion and Conclusions

Original Time Series Forecast with Auto Arima



Clearly our search for a better model led us astray, as I managed to dramatically over compensate our noise and variance reduction and over fit our series to build a forecasting model.

Despite having noise and some seasonality that we had to deal with, the most appropriate model of our initial time series was a SARIMA(1,1,1)(0,1,0)[52] model, which according to the empirical analysis and early visualizations we had done is not surprising. The ARIMA(1,1,1) component is not too far off the initial assumptions we had made, with the autoregressive component of the first order, a first order differencing term, and a moving average term of the first order which was not present in my analysis. Also, the (0,1,0)[52] falls in line with assumptions we had speculated upon earlier. I anticipated there being a seasonal attribute that repeated yearly with the seasons, and the seasonal difference of order 1 with a period of 52 weeks is that assumption.

My forecasting models were not successful in successfully capturing the patterns and rhythms of the time series of the weekly water temperature readings over the course of 2020-2022. Looking back at the models I developed, I overcompensated in trying to reduce the variability of data set. This resulted in skewed estimates and essentially useless forecasts. I did succeed, however, in developing models that were stationary with residuals that resembled white noise. Unfortunately this came at the expense of any actual utility.

The best model fit by the auto.arima() function of our initial time series data features what seems to be an upward trend, signifying that future water temperatures are expected to be increasingly warm. This can be seen directly in the plot of the ARIMA(1,1,1)(0,1,0)[52] graph, and actually is about the only thing my models may have predicted in the plots of their decomposition into data, seasonal, trend and remainder components above. This is a very interesting conclusion to take note of, as it may be evidence of the local effect of climate change and warming seas- a welcome unexpected pattern to see.

Bibliography and Appendix

National Data Buoy Center. (n.d.). NDBC - Improvements. Retrieved October 28, 2021, from https://www.ndbc.noaa.gov/improvements.shtml (https://www.ndbc.noaa.gov/improvements.shtml).