Buoy Data

Gutierrez, Mena, Thompson, Silva

11/21/2021

Abstract

This particular data set comes from the Santa Monica Bay buoy collected by the National Data Buoy Center. The objective of testing this data set is to forecast the significant wave height in meters which is calculated as the highest one-third of all the wave heights during the 30 minute sampling period. Beginning with extracting the data from the ndbc.noaa.gov website, we data cleaned our data set and modified it to fit the desired intervals of 30 minutes since the time intervals were uneven. Our desired forecast model should be able to forecast six hours, one week, and 3 months into the future. Our first attempt was to design a SARIMA model to find the best fit. However, since we suspected there was multiple seasonalities, we proceeded with the Dynammic Harmonic Regression to further expand our search. Although the DHR was more successful than SARIMA fit, none of our models performed as desired. Hence, we concluded that we should select different predictors such as wind speed or wind direction to indicate when they are correlated with the significant wave height at certain values of h for future forcasting.

Data Cleaning

Comments on our data cleaning process.

The data was pulled from the ndbc.noaa.gov website. There were many buoy locations to choose from where we could select one to pull from. We ended up getting the data by reading a url that contained a text file extension. From this extension we were able to read it into a table. Since our focus was on forecasting significant wave height we selected that column of data. The time intervals at first seemed to come in every 30 minutes.

Our next task was to organize the dates and times associated with each data point. Since the data was organized in a way that the year, month, day, hour, minute, and seconds were each individual columns, we had to use concatenation to turn them into one column. Using a date class, we were able to create a new data set with each of the dates as a date object to be matched with the corresponding data points.

Now that we had our time series data, we turned it into a tsibble object, using build_tsibble to specify the time intervals more clearly. We also took note that some of the intervals were not perfectly matched up (i.e. not exactly 30 min intervals at some points) This was most likely due to the actual buoy recording was not perfect. This was remedied by adding or subtracting 3 or 4 minutes to the times at certain sections of the data.

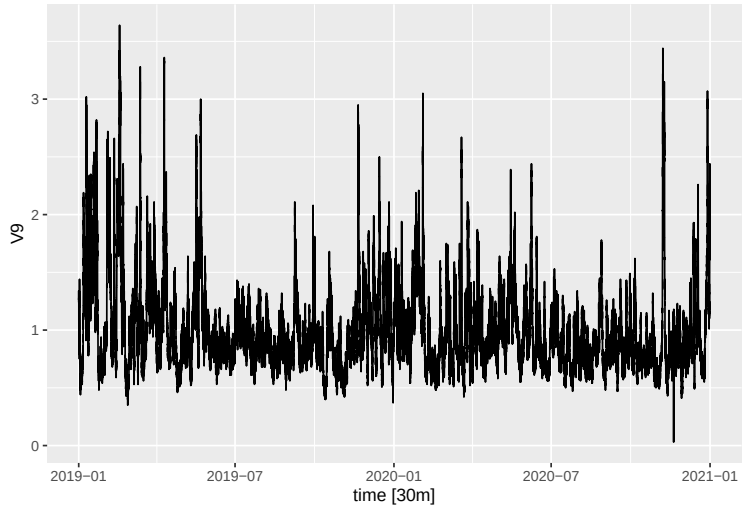
Next, we still had a few large gaps in our data due to missing data from the buoy. We used an ARIMA interpolation to fill in those missing gaps

Analysis

We started the analysis by breaking the data up into separate training and testing sets. We did all of the initial analysis and model fitting exclusively on the training set. Doing this was important to ensure that we were able to test the models we built using only the training set by checking how well they forecast into the testing set that contains known observations. Keeping the testing set quarantined from the training set during the process is crucial.

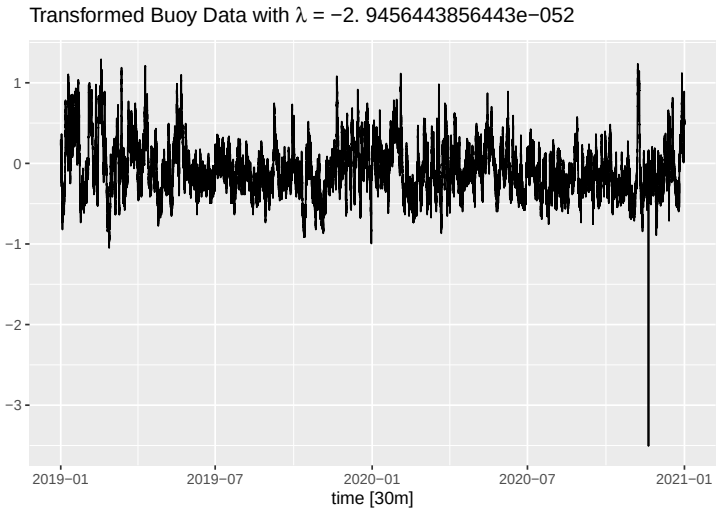
Transformations

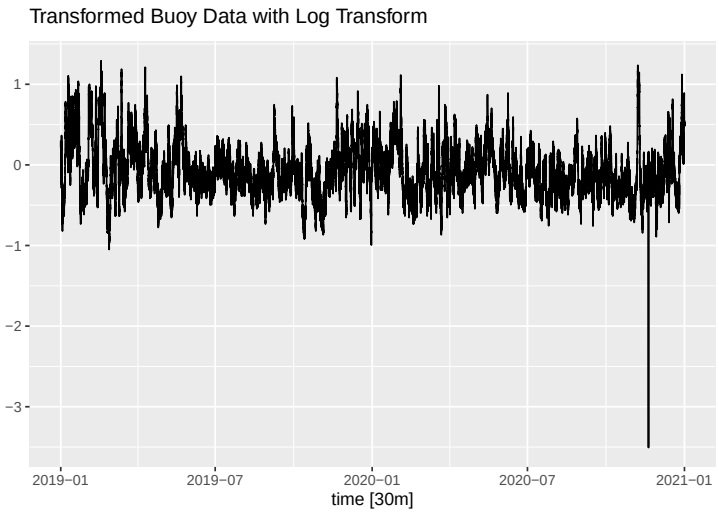
Next, we plotted the data and tried to identify trend and non-constant variance. If either one is present in the data, the data will not be stationary which would require us to make some transformations to the data before we are able to proceed with model fitting.




```
lambda <- buoy_train %>%
  features(V9, features = guerrero) %>%
  pull(lambda_guerrero) ; lambda
```

```
## [1] -2.945644e-05
```

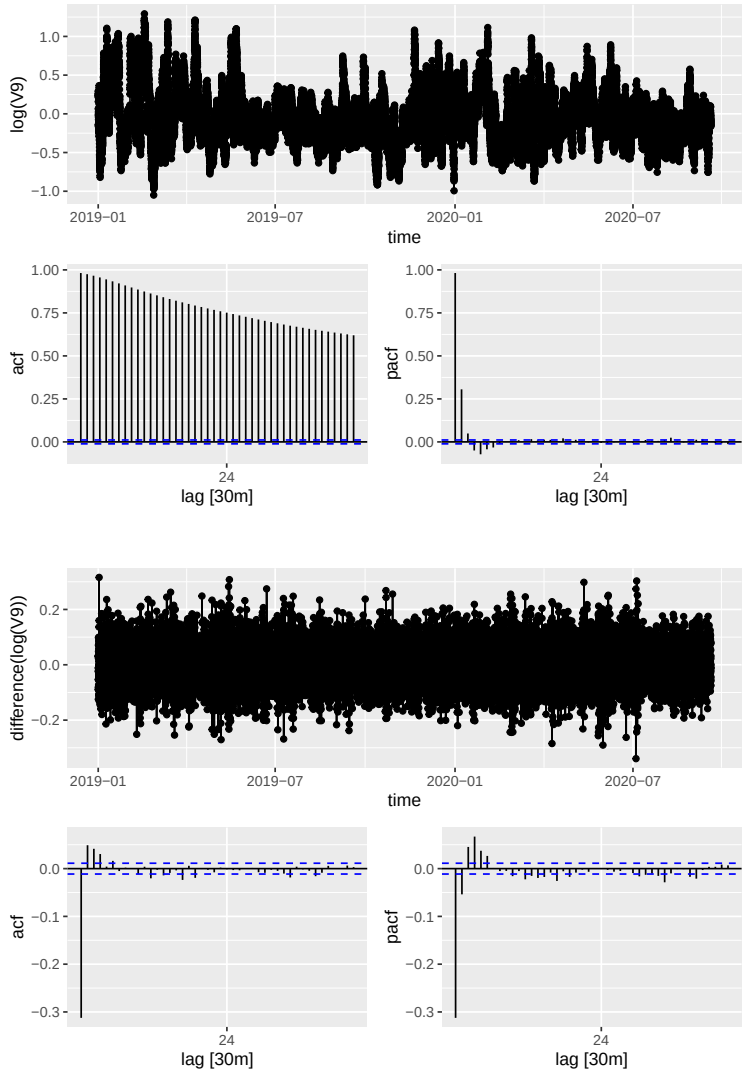




The data appears to have some non-constant variance with wildly varying magnitudes of lag heights throughout the two year period. To determine if a transformation is necessary, we performed a Box-Cox Test. The first thing we had to determine was an appropriate value for lambda to use in the Box Cox Test. For reference, $\lambda = 1$ indicates no transformation is needed; $\lambda = 0.25$ indicates a fourth root transformation; and $\lambda = 0$ indicates a log transformation is needed. Using a built in function in R, we determined that the suggest value is $\lambda = -0.00002945644$. Notice, that this value is extremely close to zero which suggests that using a simple log transformation is probably the best transformation to use. We plotted both transformations and found the two to be nearly identical. Thus, we decided to proceed with the rest analysis using a log transformation of the data.

Differencing

Once we determined that a log transformation was needed, we checked if the data needs to be differenced.



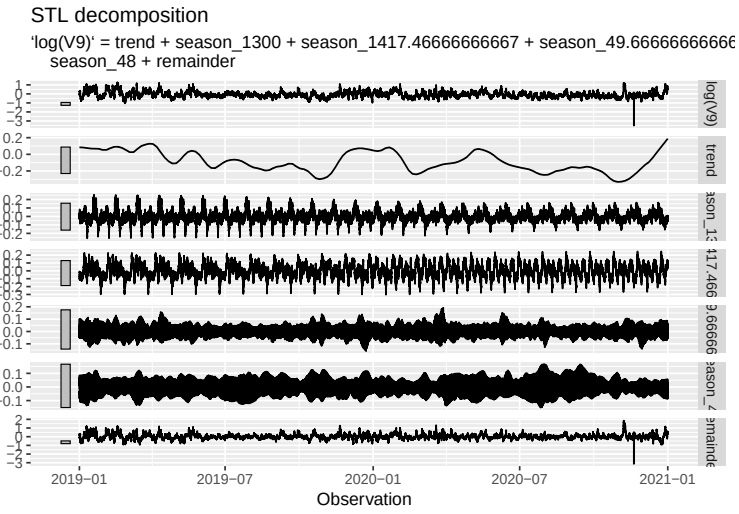

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1   0.00174         0.1
```

The series appears level overall but the lags present in the ACF plot exhibit a very slow decay which indicates the data is nonstationary and should probably be differenced. Taking a single nonseasonal difference appears to have made the data stationary since there is no longer any slow decay in the ACF plot. There are also now lags with clear cutoffs and tailing behavior which is typically helpful in the model fitting process. If we choose to use a typical SARIMA model fitting, we now know we will need to use a model that has a nonseasonal difference order of $d = 1$.

In order to determine if another differencing is required, we performed a unit root test to test the stationarity of the nonseasonal difference(log(V9)). Since the p-value = 0.1 and is larger than 0.05, it suggests that no further differencing is required. Hence, we only performed one nonseasonal difference after the log transformation.

STL Decomposition

```
##  year  week   day  hour
## 17532   336    48     2
```



Analyzing the seasonal decomposition of was difficult due to data having possibly multiple seasonalities. Determining what those multiple seasonalities was also challenging.

Based on the some knowledge of the data, it's seasonalites could be related to the periods of the sun and the moon. So trying out different lunar days, months and years in combination with solar days, months and years was returning the most significant results.

What resembled the most seasonal components were the lunar year, lunar month, lunar day, and solar day. Each seemed to contain some level of seasonality, but none were perfect.

Model fitting

Since the data seemed to contain multiple seasonalities and the seasonal periods were quite long, we choose to use Dynamic Harmonic Regression to try and fit the model. It was a method that was mentioned in the FPP3 text book and the package seemed to know how to deal with it reasonable well.

The idea behind it was to use a fourier terms to approximate the seasonal components of the model and to use an ARMA model to account for the short term dynamics.

The model looks like this: $y_t = bt + \sum_{j=1} K[\alpha_j sin(\frac{2\pi jt}{m}) + \beta_j cos(\frac{2\pi jt}{m})] + \eta_t$

Where η_t represents the ARMA process, m is the seasonal period, K is the number to fourier terms added up and bt is similar to a drift term.

Selecting the values of K were done on the basis of AICc and a function was created to test all the values over the possible seasonal periods. However, the computations were very long, so the only K values that were tested were from 1:3. This process showed that from the values tested: K=1 for lunar year, K=1 for lunar month, and K=2 for solar day.

```
## # A tibble: 4 x 8
##   .model  sigma2 log_lik     AIC    AICc     BIC ar_roots  ma_roots
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <list>    <list>
```

```
## 1 mk1    0.00336  42989. -85955. -85955. -85864. <cpl [2]> <cpl [3]>
## 2 mk2    0.00336  42995. -85947. -85947. -85773. <cpl [2]> <cpl [3]>
## 3 mk3    0.00333  43149. -86266. -86266. -86133. <cpl [4]> <cpl [2]>
## 4 mk4    0.00333  43157. -86265. -86265. -86057. <cpl [2]> <cpl [3]>
```

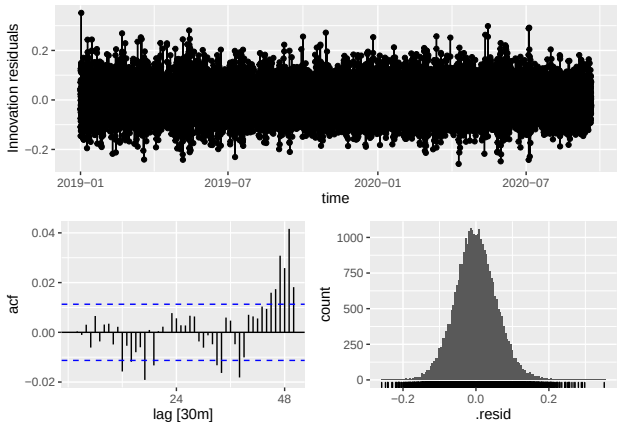
```
## # A tibble: 5 x 8
##   .model  sigma2 log_lik     AIC    AICc     BIC ar_roots   ma_roots
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <list>     <list>
## 1 m4     0.00338  42893. -85772. -85772. -85714. <cpl [17]> <cpl [19]>
## 2 auto   0.00338  42890. -85771. -85771. -85729. <cpl [1]>  <cpl [3]>
## 3 m1     0.00338  42888. -85760. -85760. -85693. <cpl [53]> <cpl [0]>
## 4 m2     0.00338  42887. -85760. -85760. -85701. <cpl [8]>  <cpl [0]>
## 5 m3     0.00341  42789. -85562. -85562. -85496. <cpl [2]>  <cpl [80]>
```

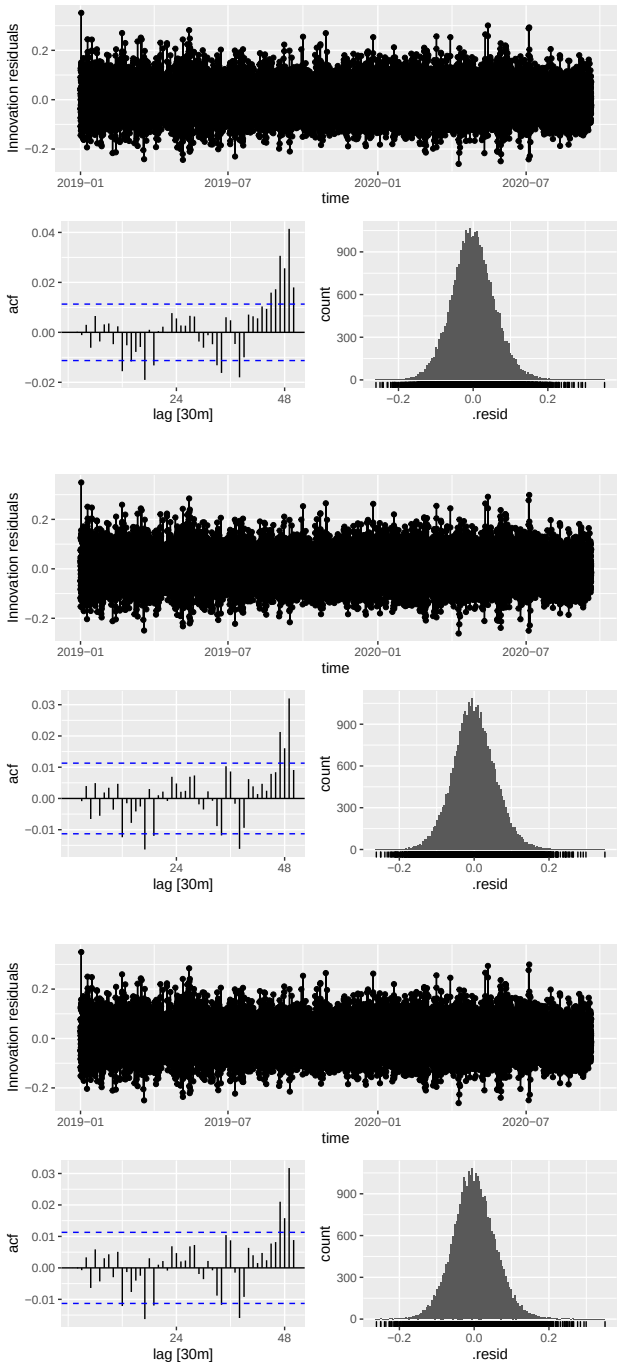
In order to compare the AICc's values across ARIMA models, we kept the differences the same where $d = 1$ and $D = 0$ since we only performed one nonseasonal difference. Within this model it was difficult to see any significant seasonal pattern and therefore we chose arbitrary numbers for 's' resulting in our choices of period 2 (1 hour), period 16 (8 hours) and period 24 (12 hours).

After observing the PACF for the nonseasonal AR component, we tested the first ARIMA model to have a nonseasonal AR(6) component because the ACF plot looks like it is tailing off and the PACF cuts off at lag 6. However, we did think that probably the PACF was cutting off at lag 2 and the rest of the lags could be part of the seasonal AR component. Therefore we also tested nonseasonal AR(4) (m2) since its the highest positive spike in the PACF, and AR(2) (m3). Similarly, we tested for the MA component by looking at the ACF and decided to keep its nonseasonal components at 0 since it looks like it is tailing off. However, for m3 we decided to assign a seasonal MA(5) with a period of 16 since the lags are decreasing.

Therefore our SARIMA models were m1 = ARIMA(5,1,0)(2,0,0)[24], m2 = ARIMA(4,1,0)(2,0,0)[2], m3 = ARIMA(2,1,0)(0,0,5)[16], m4 = ARIMA(1,1,3)(1,0,1)[16] and auto = ARIMA(1,1,3). Comparing all five SARIMA models, m4 came to be the best model obtaining the lowest AICc value of -85771.90 with SARIMA model ARIMA(1,1,3)(1,0,1)[16] where the seasonality is 16 time unit intervals of 30 minutes each (every 8 hours). Although we ran the auto ARIMA model to help us choose better models, we noticed that the output was an ARIMA model instead of a SARIMA model. This was not expected since we were testing for SARIMA models. However, we were able to beat its AICc score by obtaining a lower score, therefore we kept our m4 model as the best model.

Residual Analysis






```
## # A tibble: 4 x 3
##   .model lb_stat   lb_pvalue
##   <chr>    <dbl>       <dbl>
## 1 mk1       211. 0
## 2 mk2       208. 0
## 3 mk3       115. 0.000000475
## 4 mk4       113. 0.000000850
```

```
##   RMSE_mk1   RMSE_mk2  RMSE3_mk3  RMSE4_mk4
## 0.06002077 0.06000852 0.05970140 0.05968510
```

(i) Looking at all the residual plots, the residuals from all models (mk1, mk2, mk3, mk4) are normally distributed and all have mean zero with only outlier present on each model located on the furthest right end of the plot. Bootstrap isn't necessary in this case.

(ii) All models appear to have constant variance, so we have nothing to be alarmed of.

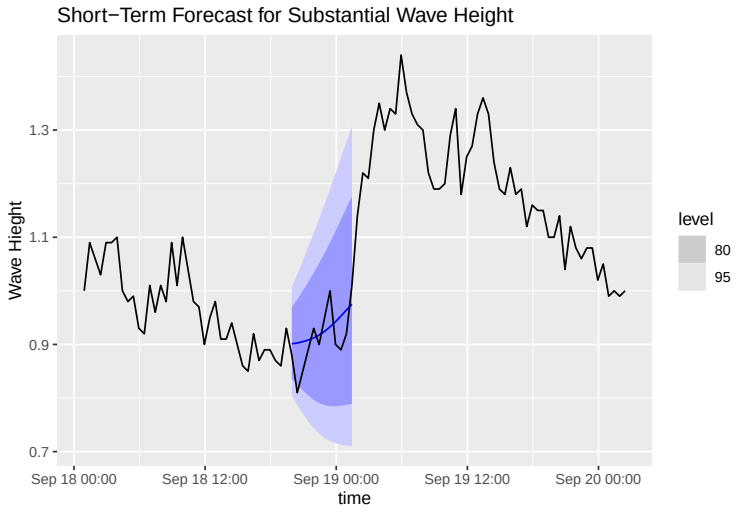
(iii) In each ACF plot we have significant lags that appear at lags $h = 12, 17, 19, 38, 46, 47, 48$. Therefore, we have some significant autocorrelation that is most apparent every 6 hours.

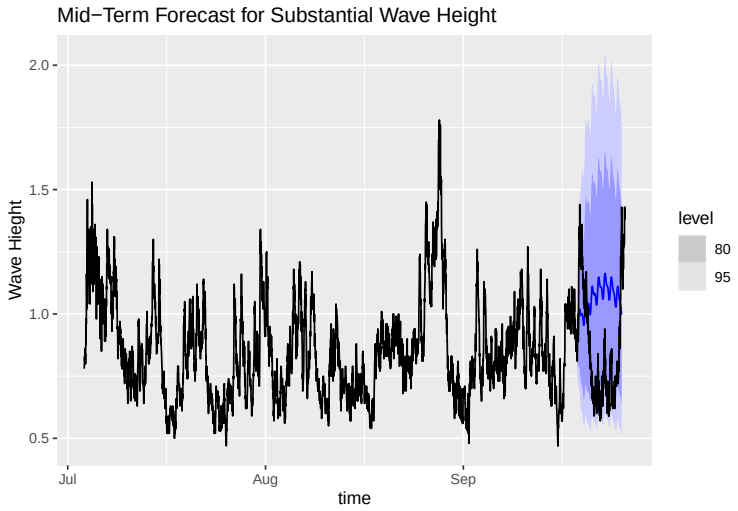
(iv) Inspecting the AICc scores, model mk4 is the best model with the lowest AICc score of -85955.16

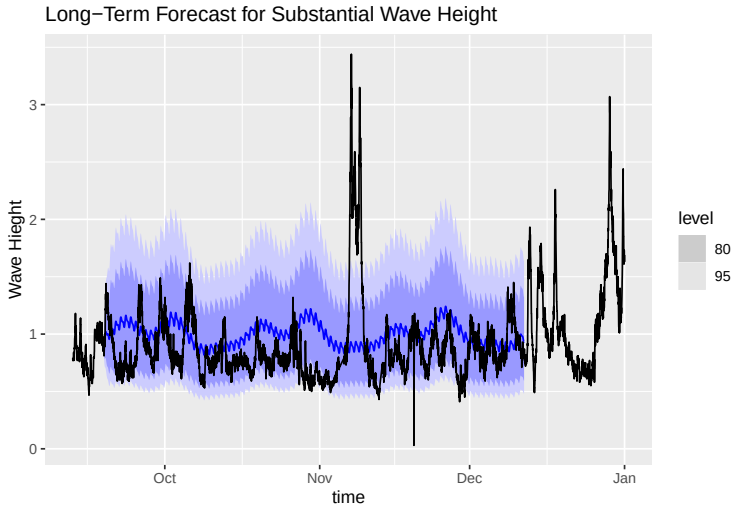
(v) The Ljung-Box test statistics all had very small p-values, meaning our residuals seem to not do well in forecasting these models. However, the model(s) that faired the best were mk1 and mk2. This means that we do not have sufficient evidence to reject our null hypothesis, that our residuals appear to be correlated and could be improved.

(vi) All RMSE were all really close within one another, however the best model with the lowest RMSE (Root Mean Squared Error) is model mk4 with an RMSE value of 0.05968510. Minimizing this value, will result in forecasts of the mean.

Forecast Testing

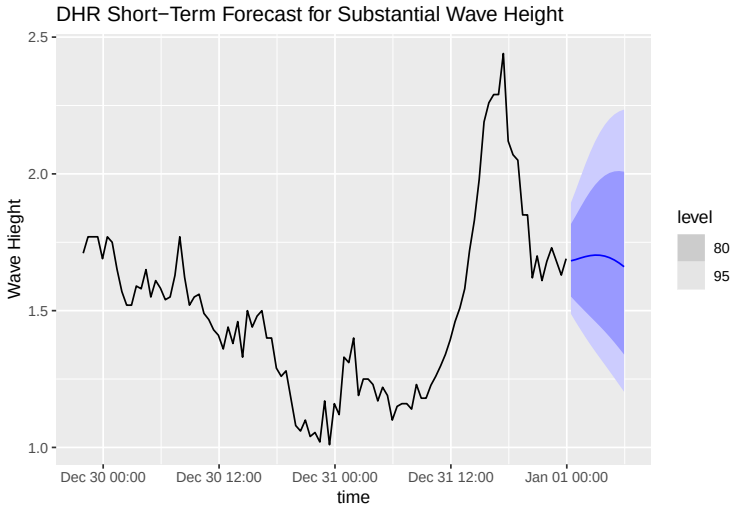


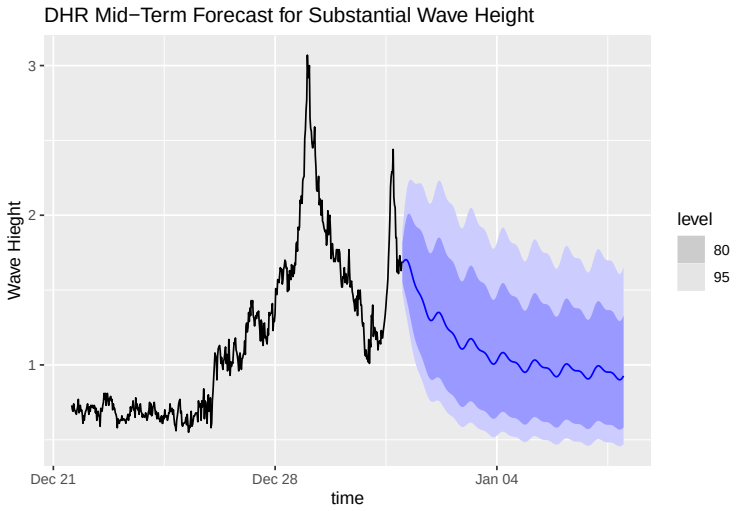


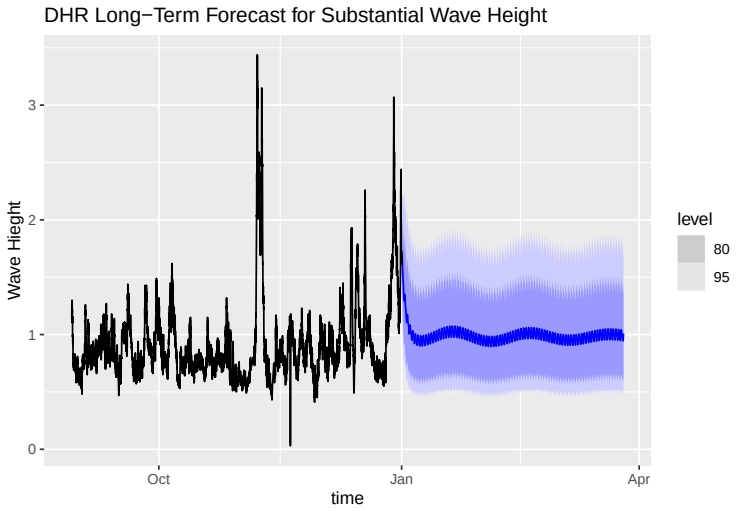


Once we identified the best model (m4), we fitted it to the training set. We then produced forecasts into the testing set and assessed it's performance by comparing its predicted values directly against the known values we kept seperately in the testing set. We produced a short-term forecast (6 hours), mid-term forecast (1 week), and long-term forecast (3 month). In these plots, we only included about a week of data leading up to the forecast to give us a better scale for comparing the forecast against the testing set. The short-term is difficult to analyze since there are so few observations in the forecasts. All of the forecast lengths have an issue with the confidence intervals being too wide. We expect to have only 95% of the data points in the confindence intervals so it is problematic when they are so wide they cover all data points. The long-term forecast is the only one to have data points outside of the confidence intervals. Overall, the mid-term appears to forecast predicted values reasonably compared to the actual data points.

Forecasting







For our actual forecast, we fit the model to the full data set including both the training and test sets and then forecasted into the future. The short-term appears to be reasonable only up to a couple of hours and the mid-term forecast appears to be reasonable for a few days. The problem is that their confidence bands start wide and stay wide so we really can't narrow down a point where the forecast becomes unreasonable. The long-term forecasts appears to really become unreliable after the the first week or so. Overall, this model doesn't seem to provide an accurate forecast with reasonable confidence intervals.

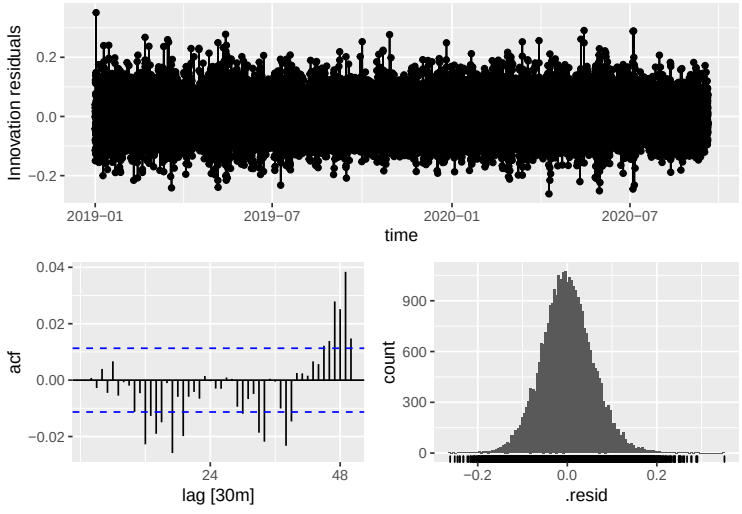
Alternatives

We spent the majority of the project attempting create a forecast for this data using Dynammic Harmonic Regression. We ambitiously attempted to capture the multiple seasonalities within this data. Unfortunately, this proved to be very complicated and computationally taxing. We decided to try forecast using a SARIMA model. We were able to come up with a model that had a lower AICc than the model that the Hydman-Kandakar Algorithym suggested.

```
glance(sarima_fit)%>%arrange(AICc)
```

```
## # A tibble: 5 x 8
##   .model  sigma2 log_lik     AIC    AICc     BIC ar_roots   ma_roots
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <list>     <list>
## 1 m4     0.00338  42893. -85772. -85772. -85714. <cpl [17]> <cpl [19]>
## 2 auto   0.00338  42890. -85771. -85771. -85729. <cpl [1]>  <cpl [3]>
## 3 m1     0.00338  42888. -85760. -85760. -85693. <cpl [53]> <cpl [0]>
## 4 m2     0.00338  42887. -85760. -85760. -85701. <cpl [8]>  <cpl [0]>
## 5 m3     0.00341  42789. -85562. -85562. -85496. <cpl [2]>  <cpl [80]>
```

Using this model we did a quick residual analysis.




```
## # A tibble: 5 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 auto      254.         0
## 2 m1        257.         0
## 3 m2        269.         0
```

```
## 4 m3        529.         0
## 5 m4        249.         0
```

```
##    RMSE_m1    RMSE_m2   RMSE3_m3   RMSE4_m4
## 0.06022162 0.06022367 0.06041953 0.06021139
```

(i) Now observing the count plots for models auto, m1, m2, m3, m4, the residuals on all models are normally distributed, with only a few outliers apparent, which we can assume that it will not affect our forecasts.

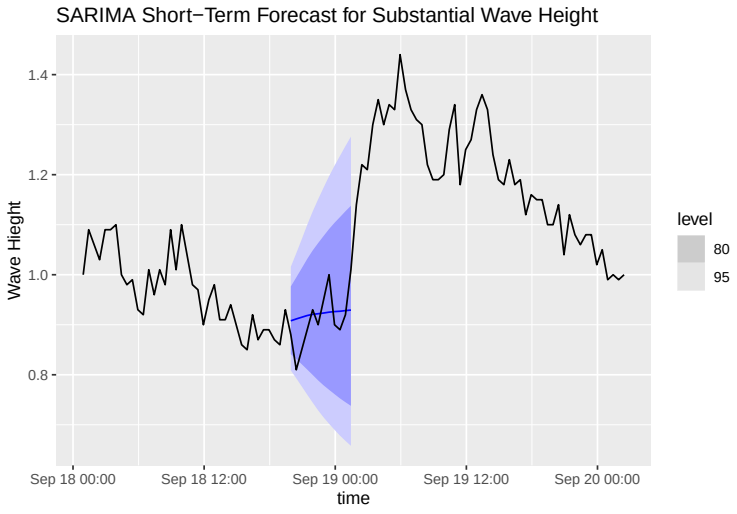
(ii) Models auto, and m1 - m4 appear to also have constant variance, we do see evenly spaced out spikes, which could be due to seasonality in our time series.

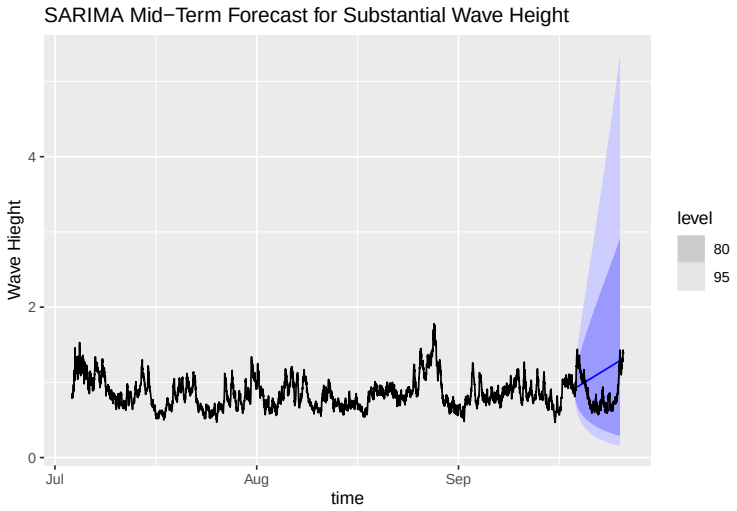
(iii) In models auto and m1 - m4, their ACF plots also express the same behavior where we have several significant lags where the most common significant lags are $h = 46, 47, 48$ for all models. We do have one exception, model m3 has its most significant lags at $h = 3, 4$. Henceforth, our residuals in all our models appear to be correlated.

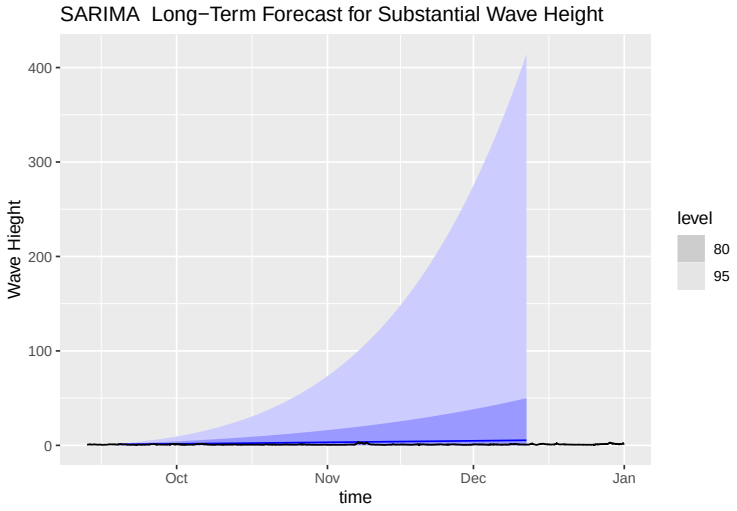
(iv) Out of our SARIMA models, model m4 is the best with an AICc score of -85771.90.

(v) For our SARIMA models, all models m1 - m4 have infinitecimally small numbers, therefore we can not accept our null hypothesis and therefore do not have sufficient evidence that our residuals appear to be correlated.

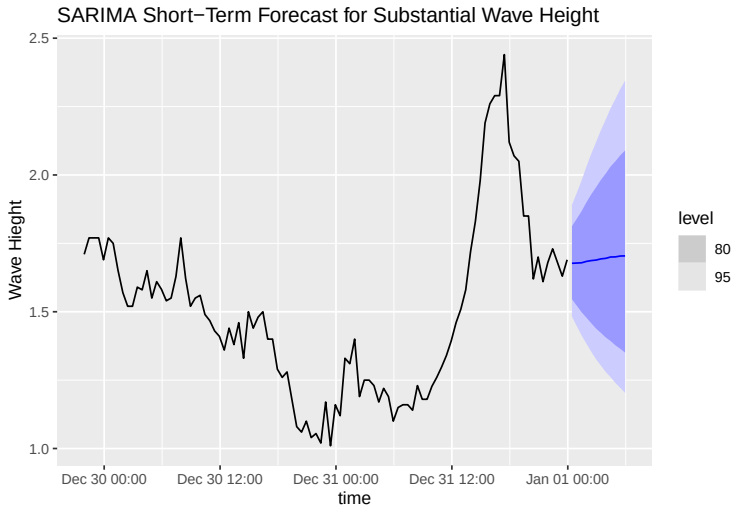
(vi) All RMSE values for models auto - m4 were also close within one another, but if we were to pick, we would choose the smallest values, which is models m4, giving an RMSE value of 0.06021139.

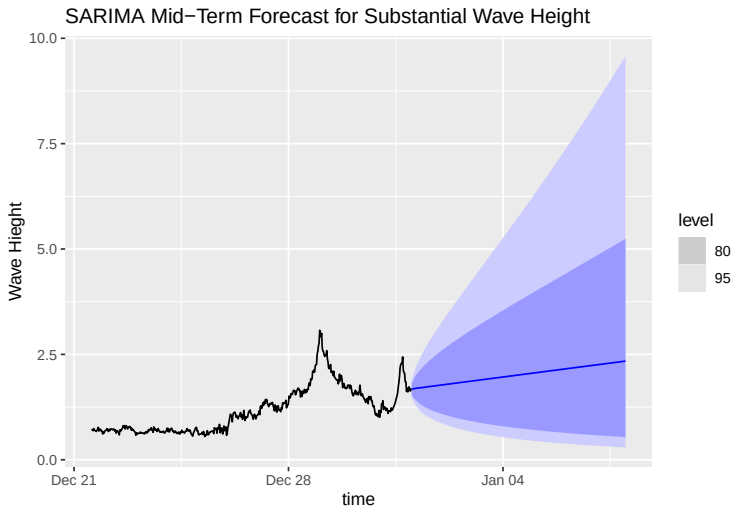


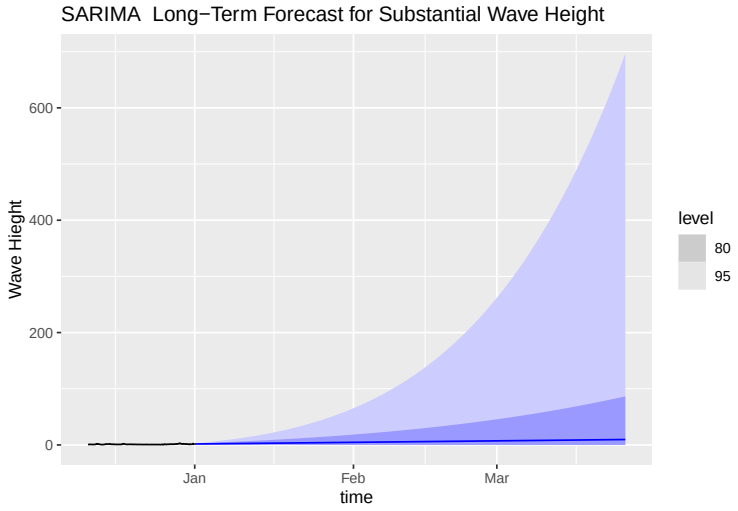




Overall, all three of the forecasts did not perform any better that the forecasts we made using dynamic harmonic regression. The short term does not even appear on the plot at all since there are only handful of observations in the short 6 hour forecast window. The mid-term forecast performs the best out these three since its projected points line up reasonably with the actual data in the test set. The long-term forecast completely unreliable. The confidence intervals shoot up exponentially which changes the scale of the plot making the data appear as a flattened line. Since this model did such a poor job of forecasting the test set, there really no point in refitting this model to the full data set to attempt an actual forecast beyond the test set.


SARIMA Short−Term Forecast for Substantial Wave Height
Wave Hieght
2.5
2.0
1.5
1.0
Dec 30 00:00
Dec 30 12:00
Dec 31 00:00
Dec 31 12:00
Jan 01 00:00
time
level
80
95


SARIMA Mid−Term Forecast for Substantial Wave Height
Wave Hieght
10.0
7.5
5.0
2.5
0.0
Dec 21
Dec 28
Jan 04
time
level
80
95


SARIMA Long−Term Forecast for Substantial Wave Height
Wave Hieght
600
400
200
0
Jan
Feb
Mar
time
level
80
95

Future Considerations

Seeing as the attempted models did not do a great job of forecasting the future values, incorporating predictors into the Dynamic Harmonic Regression model would be an interesting task. Since, we propose that wind, tide or other variables might have an effect on the significant wave height. There are numerous other predictor variables that could have an correlation with the our forecasted variable. Due to time and the scope of this project, they were not considered, but could potentially have positive results. This project has opened our eyes to the many other possibilities that could be done if forecasting time series.