Thad Hoskins
Gina McFarland
Paul O'Leary

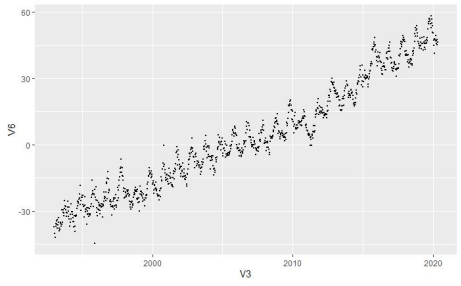
COMP 4441
20 Aug 2020

Prediction of Global Mean Sea Levels Using Three Statistical Methods

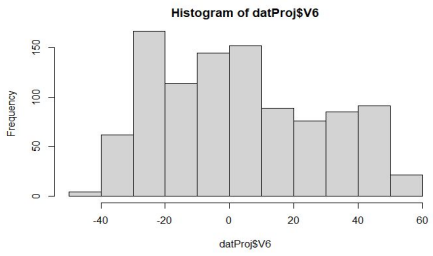
Rising ocean levels are a frequent topic in the news, but how quickly are they rising? Is this occurring at a predictable rate? This is what our team set out to analyze using data collected by NASA. We analyzed the data set spanning 27 years using three statistical methods: linear regression, time series analysis, and seasonal naïve forecasting. Specifically, we sought to answer the question, if we subset the dataset, can we create a model based on early data that successfully models later data values?

Data

The data contains Global Mean Sea Level (GMSL) variations computed at the NASA Goddard Space Flight Center that was collected as a part of the Making Earth System Data Records for Use in Research Environments (MEaSUREs) Program. In this program, NASA uses satellite sensors to gather data on the oceans, land, atmosphere, ecosystems, and the interactions of all of these (2020). The data is publicly available at PODAAC Drive. There are 12 columns in the data. Columns 1 - 5 have information on how the data was collected, Col 6 contains the raw data, and columns 7-12 have varying analyses on the raw data. We concluded that it was best to study the raw data as opposed to data that has had any sort of manipulation or filtering for this analysis. The units for the raw data in column 6 are presented in terms of global mean sea level (GMSL) (Global Isostatic Adjustment (GIA) not applied) variation in (mm) with respect to 20-year TOPEX/Jason collinear mean reference. The TOPEX/Jason collinear mean reference is considered zero and there are positive and negative measurements in the data with respect to that point.



Plot 1

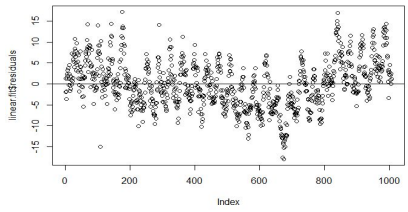


Plot 2

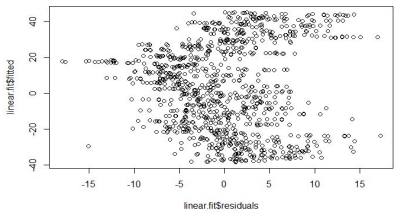
The data appear to be slightly skewed right. The data had no missing values in the variables that we researched and there were no outliers. For the Time Series and Seasonal Naïve we did have to adjust the time value slightly. We removed the points for the partial year, 2020. The observations were taken approximately 3 times per month over the 27 years included. The measurements are taken close to the same day each time, within a day or two, but the time series methods require exact intervals. We decided since the actual measurements are so close to being equal intervals (an average difference of 0.02716 days), 994 observations remaining after we removed the 2020 data, that it was reasonable to make the time intervals equal in our analysis.

Linear Regression

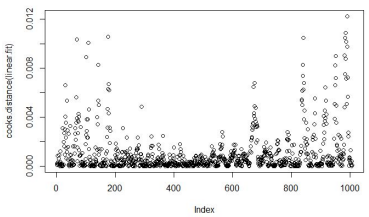
Our first approach to predict the data was linear regression. The data has a clear linear relationship based on Plot 1. In linear regression, a relationship between two variables is modeled by fitting a linear equation to the points in the observed data (1997). This model is used to predict an outcome based on a predictor variable. In order for this method of prediction to work, several assumptions about the data must be true. First, there must be a linear relationship, the observations must be independent, the response variable must be normally distributed, no auto-correlation, and homoskedasticity, that is, the variance of the residuals must be the same for any X.



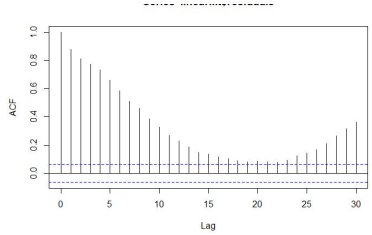
Plot 3



Plot 4



Plot 5



Plot 6

When we tested the model to ensure the assumptions held, there were several issues. In Plot 3, the residuals of the function do not appear randomly distributed as is required for linear regression to be used to create inferences. This implies that the variables are not independent. Plot 4 is the fitted vs. residuals plot, which shows non-linearity, unequal error variances, and outliers. For the linear regression model assumptions to hold valid, this plot should have a horizontal band that is distributed evenly around the 0 line with no one residual standing out from the others. This is not the behavior shown in the plot with the points grouped and with several outliers. Plot 5 is a plot of Cook's distances. These are influential points that may affect prediction outcomes. This data has a number of points with high Cook's distances. Finally, Plot 6 shows autocorrelations. Autocorrelations occur when the residuals are not independent of each other, which is a violation of the assumptions of the linear regression model. Since this plot shows a number of points above the blue line, there are autocorrelations in the data. Based on these issues, we are unable to model this data with linear regression.

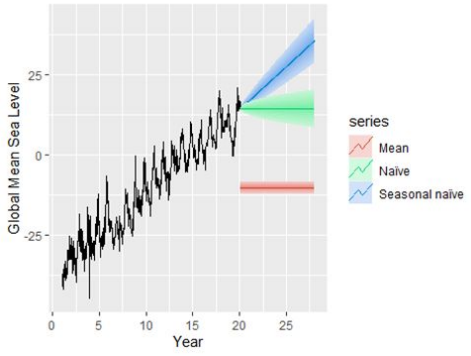
Seasonal Naïve Forecasting

Our next approach to model the data was Seasonal Naïve Forecasting. This is a type of Time Series forecasting. Naïve Forecasting makes predictions based on the idea that future data will look like past data. It uses a baseline assumption that the data is relatively unchanging. Seasonal Naïve Forecasting assumes the data changes seasonally or periodically in similar ways, e.g., Christmas sales or Summer ice cream sales. In those examples, the season of Christmas or Summer have higher sales, therefore the forecast adjusts during those seasons in the forecast. Further, we can add a "drift" to the seasonally Naïve forecast to allow for a slight increase or decrease, giving us a Random Walk Model.

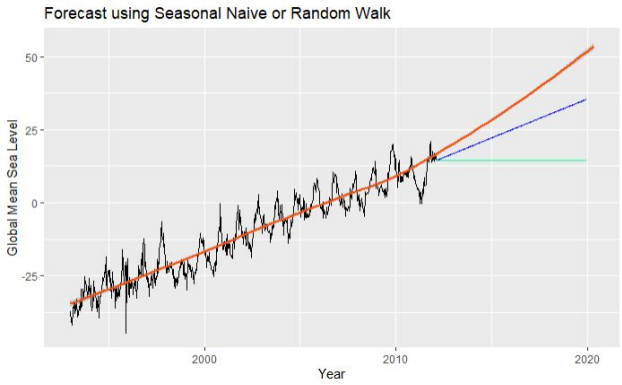
The Random Walk Model seemed particularly ideal for this dataset. When analyzed, the data behaves in a seasonal up and down pattern and slowly increases, i.e., drift.

In R, similarly to the other methods and models, we took the first 70% of the data to train the model.

```
drift.proj <- rwf(train.clean, h=length(ts.test), drift=TRUE, level=c(1:10))
naive.proj <- naive(train.clean, h=length(ts.test), level=c(1:10))
```

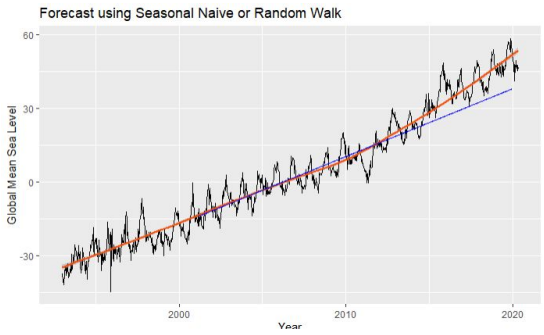


With the algorithm and graphs behaving unexpectedly, though pointing in the right direction, we compared the indicated trend to the actual trend.



As this clearly shows, the forecast predicts the prior 19 year trend continuing. The actual data indicates a slight upturn in the trend which departs from the forecast almost immediately. Imagining bad luck in our chosen training data cutoff, we retrained using an earlier data, 2001 instead of 2012. The result was dramatic.

```
ts.train <- ts(ts.dat[1:303], frequency = 36.8)
```



The model now predicts with a high degree of accuracy the next 11 years, departing at our previously chosen point in 2012.

The result of this model is clear in the description. Naive (seasonal, random walk, or original) assumes a continuing trend. When the trend continues, as seen in the final image, the prediction may be accurate. When the trend changes, the Naive forecast shows the departure, hence its classification as a baseline.

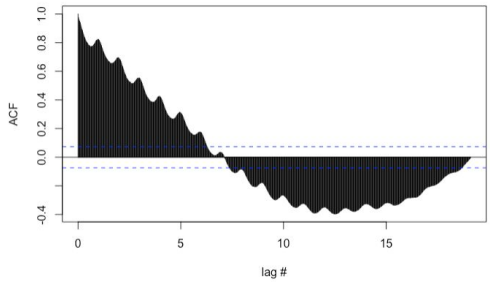
Time Series Analysis

ARIMA Modeling

ARIMA (autoregressive integrated moving average) is another commonly used modeling technique for forecasting time series data. Certain aspects of time series data must be analyzed before ARIMA can be completed.

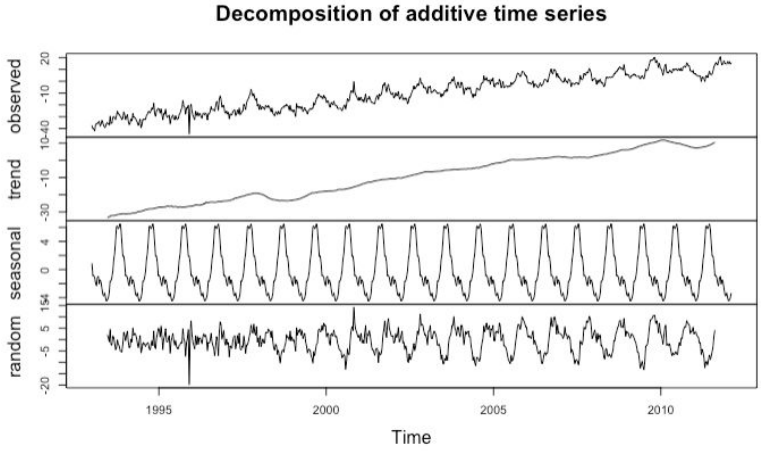
First, determine whether the data is additive or multiplicative. Please recall the original data plot. Consider the difference between the minimum value and the maximum value for one year worth of data in this graph, 2000 for example - that difference will be close to the min/max difference for the year 2018. This would not be the case for a multiplicative time series - This series is additive.

Next, determine if the data is stationary or not. The ARIMA forecasting model requires that the data be stationary. The Augmented Dickey-Fuller Test or an Autocorrelation test can be performed to determine if the data is stationary. The resulting plot of an autocorrelation test is shown here:



R Command: acf(trainTS,lag.max = length(trainTS), xlab = "lag #", ylab = 'ACF', main='Autocorrelation, GMSL Data')
This test is correlating current time steps with previous time steps, and showing the “lag”. Because the mean of the data changes over time, this data is not stationary, and will have to be manipulated before running ARIMA modeling.

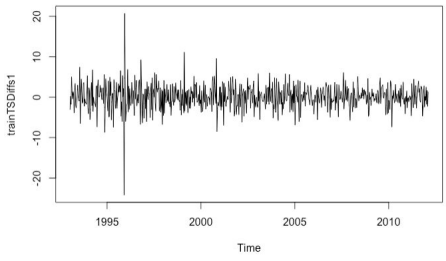
Determine if the data displays a seasonal component. The data in this time series appears to be seasonal. We can verify this by using the “decompose” command from R. Running decompose on the time series data results in this set of plots:



R Commands: decTrainTS <- decompose(trainTS, type = "additive") plot(decTrainTS)

This decomposition of the data shows plots of the original data, the overall trend, the seasonal component, and the remainder or “noise” component. Decompose uses a “moving average” to determine the trend. The moving average removed, revealing the seasonal component. When that is removed, everything leftover is the remainder or the noise in the data. Based on the steadiness of the seasonal component, seasonality is confirmed.

The ARIMA model requires that data be stationary. Non-stationary data can be “differenced” to convert it to stationary data, using the “diff” command in R:

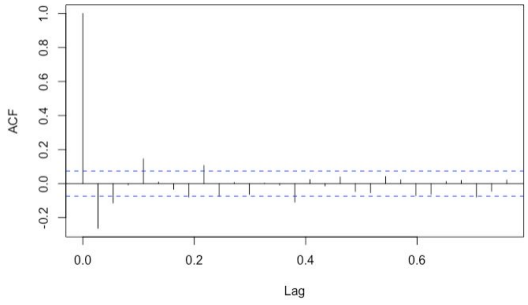


R Commands: trainTSDiffs1 <- diff(trainTS, differences = 1) plot.ts(trainTSDiffs1)

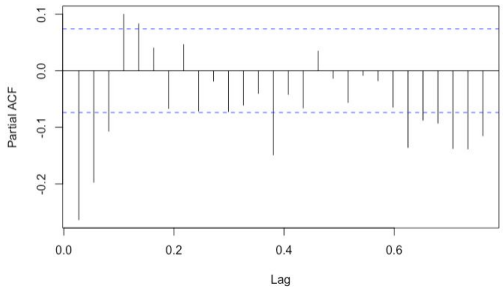
Some data will need to be “differenced” multiple times to convert it to stationary data. This number of differences, *d*, will be used by the ARIMA function.

ARIMA(p, d, q) (Autoregressive Integrated Moving Average) is a generalized version of ARMA which is based on AR, auto regressions and MA, moving average data. In addition to

the number of differences, *d*, ARIMA has a *p* and a *q* component, which can be estimated with ACF, Autocorrelation and PACF, Partial autocorrelation plots. The following ACF and PACF plots were run on the once differenced data:



R Command: datCorr <- acf(trainTSDiffs1)



R Command: datCorrPartial <- pacf(trainTSDiffs1)

p can be estimated from the PACF plot, counting from left to right the number of lags until the data trends to 0, and within the confidence interval. *q* can be estimated from the ACF plot in the same manner. Running differencing more times, resulting in a different value for *d* will affect subsequent estimates for *p* and *q*. Numerous attempts were made in this manner, and ARIMA(p, d, q) was run each time. ARIMA creates AICc and BIC (Akaike and Bayesian Information Criteria) estimates of the quality of the trial run to help determine the best model. However, none of the models attempted yielded any meaningful results.

Fortuitously, R has a built in function called “auto.arima” which performs all of the tests described above, and returns an estimate of the best possible model, given the data. The auto.arima command returned entries for a Seasonal ARIMA model, which adds (P, D, Q) components for the seasonal portion of the data. Here are the results of that command:

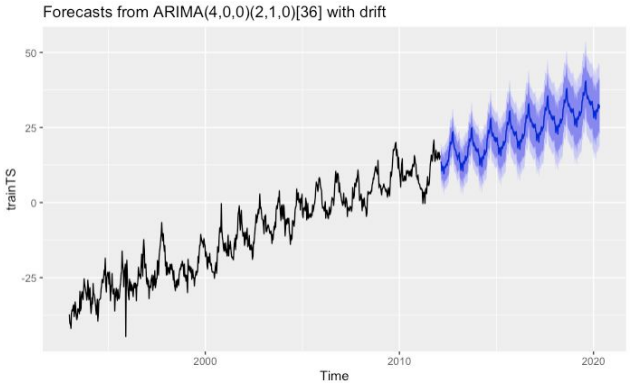
R Command: auto.arima(trainTS, allowdrift = TRUE, stationary = FALSE, seasonal = TRUE, method = "ML")

Returned:

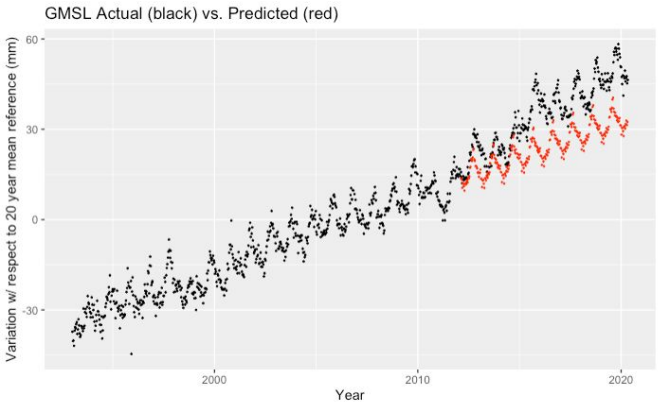
```
Series: trainTS
ARIMA(4,0,0)(2,1,0)[36] with drift
...
```

These estimates were run and plotted with:

R Command: arimaResults <- Arima(trainTS, order = c(4,0,0), seasonal = c(2,1,0), method = "ML", include.drift = TRUE)
arimaResults %>% forecast(h = length(testTS)) %>% autoplot()



Finally, here is a plot of the predicted data overlaid on the original plot of all the data:



Test Results:

A number of tests were run to check the "goodness" of the predicted results:

MAE (Mean Absolute Error): 11.083. The mean of the differences between actual and predicted data points is 11.083 units (mm). For some data, the relative size of this error may not be obvious. In this case, 11.083 is a large error.

MAPE (Mean Absolute Percentage Error): 0.299. The mean of the differences between actual and predicted data points expressed as a percentage. This value shows that the predicted points are off by a mean value of about 30% from the actual data points.

R-Squared: 0.759. R-Squared is the percentage of the variation between predicted and actual points explained by a linear model. Values are always between 0 and 100%. In general, the higher the R-Squared value, the closer the model predicts the data.

Conclusions:

A model is only as good as the training data from which it is built. The Random Walk and Seasonal ARIMA models seemed to be good matches for the data, but when overlaid with the actual data, it is obvious that both models are not perfect. However, as the Seasonal Naive Forecast points out above, the data set just happened to have a change in rate of increase very near to the chosen 70% / 30% split. The Seasonal Arima model created what would have been a very good model, had the data continued with the trend from the training data.

According to a 2019 National Geographic article, the ocean has risen 8 inches since 1880 to today, with 3 of those inches since 1994. A 5 inch increase from 1880-1994 is a rate of 0.044 inches/year, but a 3 inch increase in the 25 years since 1994 is about 0.12 inches/year. This suggests that the yearly increase in mean sea levels may continue to grow at a faster rate.

References:

Coghlan, Avril. A Little Book of R For Time Series. Retrieved from https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/index.html

GSFC. 2017. Global Mean Sea Level Trend from Integrated Multi-Mission Ocean Altimeters TOPEX/Poseidon, Jason-1, OSTM/Jason-2 Version 4.2 Ver. 4.2 PO.DAAC, CA, USA. Dataset accessed 2020-08-08 at http://dx.doi.org/10.5067/GMSLM-TJ42

Hyndman, Rob J, Athanasopoulos, George. (2018) Retrieved from https://otexts.com/fpp2/

Linear Regression. (1997). Retrieved August 15, 2020, from http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

Making Earth System Data Records for Use in Research Environments (MEaSUREs) Program. (2020, February 18). Retrieved August 15, 2020, from https://earthdata.nasa.gov/esds/competitiv-programs/measures

Nunez, C. (2019, February 27). Sea level rise, explained. Retrieved August 18, 2020, from https://www.nationalgeographic.com/environment/global-warming/sea-level-rise/

Piexeiro, Marco. (2019). The Complete Guide to Time Series Analysis and Forecasting. Retrieved from https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775

United Nations. (2017, May). Retrieved August 17, 2020, from https://www.un.org/sustainabledevelopment/wp-content/uploads/2017/05/Ocean-fact-sheet-package.pdf