**Time Series for Strategic Business Forecasting and Insights**

STA 9701

Sierra Levy

A graph of a graph

Description automatically generated with medium confidence12/19/2023

Table of Contents

[**Introduction** 3](#_Toc153867124)

[**1: Data & Transformations** 3](#_Toc153867125)

[1. A. Stock Data: 3](#_Toc153867126)

[1. B. Additional Datasets: 6](#_Toc153867127)

[Citi Bike Dataset 6](#_Toc153867128)

[NOAA Ocean Anomaly 8](#_Toc153867129)

[**2: Threshold Models – Chapter 15 Analysis** 10](#_Toc153867130)

[2.1. Graphically Exploring Nonlinearity 10](#_Toc153867131)

[2.2. Tests for Nonlinearity 12](#_Toc153867132)

[2.3. Polynomial Models 14](#_Toc153867133)

[2.4. First Order Threshold ARIMA / Higher Order Threshold 15](#_Toc153867134)

[2.6. Testing for Threshold Nonlinearity 17](#_Toc153867135)

[2.7. Estimation of a TAR Model 17](#_Toc153867136)

[2.8. Model Diagnostics 19](#_Toc153867137)

[2.9. Prediction 19](#_Toc153867138)

[**3: ARIMA Models** 20](#_Toc153867139)

[3.A KSS Model 20](#_Toc153867140)

[3.B WDC Model 23](#_Toc153867141)

[3.C Citi Bike Model 27](#_Toc153867142)

[3.D Ocean Anomaly Model 31](#_Toc153867143)

[**3: Conclusion** 34](#_Toc153867144)

# **Introduction**

In this project I am exploring time series analysis, leveraging 4 different datasets in hopes of uncovering trends, patterns, and or seasonality within the data. The goal is to successfully build out four different ARIMA models and deliver useful insights or recommendations regarding their practical applicability in a business context. Lastly, I will walk through Chapter 15, which explores the intricacies of threshold models, and apply learnings to my personal datasets.

# **1: Data & Transformations**

## A. Stock Data:

For the first part of this project, I will walk through the initial tools and analysis used when exploring time series data. I will start with both of my stock datasets: Kohls Corp (KSS) and the Western Digital Corporation (WDC). Kohl’s Corp is a major American retail company, with over 1.7K locations across the United States. Western Digital Corporation is an American computer storage company founded in 1970. Both datasets came with daily open, close, high, low, adjusted close, and volume data. To better use the data, I am creating two new values that I will be working with: the Daily Range and Daily Returns.

* 1. Daily Range: The first transformation I created was calculating the daily price range. To do this, I subtracted the Low stock price from the High stock price for each day. This clarifies how volatile a stock is. The larger the number, the more volatile the stock is, and vice versa.
  2. Daily Returns: To calculate daily returns, I take the Adjusted Difference and divide it by the Adjusted Close price. This transformation shows me the changes in price relative to the prior day. I then turn the daily return into a percentage. This will make the data easier to interpret. In the financial industry, looking at a daily return value where the return has increased 10% is easier to read versus looking at .1.

A close up of words

Description automatically generated The first step is to examine the daily returns to help gauge stocks variability and risk. By observing the daily returns, I can see clear patterns of volatility which is critical to identify for risk assessment[[1]](#footnote-1). Below I have plotted the daily returns for KSS and WDC. There are noticeable patterns where most of the variation is consistent in the middle, but it’s the beginning and end of the series where the most extreme fluctuations occur. One of the most noticeable bursts of variance occurred in 2020, where both companies saw their lowest return on March 13, after the announcement of Covid-19. While both stocks have some similarity in their appearance, their risks aren’t identical. While variances for WDC and KSS look similar, I can see that WDC has a slightly higher return variance, which means their returns fluctuate more. This is also evident in the daily return plots. Overall, these plots help understand the typical behavior of the stock prices and can help assess risk.

A graph of red dots

Description automatically generated A graph of red dots

Description automatically generated

Next I will explore the daily range as it is a very important measure in financial analysis. The daily range is useful for identifying volatility patterns and risk measurement[[2]](#footnote-2). Initially, when plotting the daily range, patterns weren't immediately obvious. For WDC, there's a clear rightward skew. KSS presents a dense cluster of data points around the lower end of the range, with a slight rightward skew as well. Both plots also show some volatility and outliers. To reduce skewness, outliers, and volatility, I took the log values of the daily range. After plotting the log values, skewness has been reduced, both plots have their volatility minimized, and the outliers have been reduced. Specifically for KSS, I can better see autocorrelation in the data as the pattern is more visible to the eye.

A blue dotted line with white text

Description automatically generatedA graph of blue dots

Description automatically generated

A graph of blue dots

Description automatically generated A graph of blue dots

Description automatically generated

Lastly I explored the lagged scatterplot of the daily range. This is an important tool in time series analysis because it not only visualizes the relationship between two values, but it also quantifies it. Below I can see that for both KSS and WDC, they both exhbit positive autocorrelation. This is important because it lets investors know about the consistency of volatility over time. If there is autcorrelation, this means that days close together exhbit similar behaviors.

A black and white image of a pile of dots

Description automatically generated A black and white image of a scatter

Description automatically generated

However, despite the autocorrelation presented in the daily range, stock prices are unpredictable and follow no distinct trend. This is known in the financial industry as Efficient Market Hypothesis. Efficient Market Hypothesis states that historical prices are not useful in predicting future prices[[3]](#footnote-3). This means that stocks follow a random walk in the long term, despite exhibiting short term autocorrelation. The random walk nature of stock prices will present some prediction challenges since we cannot rely on past data movement.

## B. Additional Datasets:

### Citi Bike Dataset

The first dataset I selected was the NYC Citi Bike Trips from NYC Open Data. Citi Bike is a privately owned bicycle sharing service in New York City. Their data includes both ride and station data. For this project, I will be working with the daily bike ride data. The data is provided by Citi Bike themselves, starting from 2013, when the program first launched, all the way to the most recent month of 2023. For this project I decided to only pull from the years 2018-2023 for the sake of time, memory, and space on my computer (each dataset has over 1M rows, and the data was provided monthly). Additionally, I am incorporating NYC Temperature and Rain data provided by Iowa State University Website[[4]](#footnote-4). I have incorporated average daily temperature, highest temperature, lowest temperature, and total rainfall. I believe this might be useful for ARIMA modeling.

A graph of a bike

Description automatically generatedA graph of blue dots

Description automatically generated

After plotting the data, seasonal trends immediately stand out. To improve interpretability, I am going to take the log value of the daily bike rides. This should help even out the data’s spread and make the underlying patterns stand out more. The resulting log-transformed plot helps me identify the peaks and dips in the data, which seem to correspond to the times of year when ridership hits its peak and when it dips, which I would assume corresponds with warmer and cooler months. In addition to the clear seasonality, I notice an upward trend as well. This upward trend in ridership can be supported by the changes that occurred during Covid-19. While Covid-19 may have impacted public transit ridership, Citi Bike only saw an initial dip in rides during March 2020, with ridership returning to normal levels that following summer[[5]](#footnote-5). Additionally, it has been reported that New Yorkers started turning to bicycles for commuting, and the expansion of Citi Bike played an important role in that change[[6]](#footnote-6).

A graph of black dots

Description automatically generatedThe next aspect I will look at is the autocorrelation of the data. After plotting the lagged scatterplot of Citi Bike trips, I noticed there was a strong positive correlation between the number of city bike trips and the preceding day. This suggests that days with a high number of trips are most likely followed by days with similar volume. While there are some outliers in the data, this could be due to weather, like sever rain, wind, snow, etc., which would prevent people from using the service.

A number and numbers on a white background

Description automatically generatedLastly, I will explore the stability of the dataset. Given the seasonality and presence of regular patterns, it is crucial I explore seasonal decomposition as a precursor to my time series analysis. While there is an increase in the volume of users, which is indicated by the upward trend of the data, if the lowest and highest points are consistent over time, that should be a good sign. This is something I will explore more before building the model.

### NOAA Ocean Anomaly

The second dataset I picked is from the NOAA looking at Ocean Anomalies over time. The NOAA is a science based federal agency that studies various types of climate data. For this project I am looking at ocean anomalies, which are the deviations in ocean temperature from a long-term average period. The data is created by combining sea surface temperatures with land surface air temperatures. By looking at ocean anomalies we can have a better understanding of climate change as it helps identify shifts in global patterns. I selected monthly data dating from January 1900 to October 2023, giving me 1,486 rows of data. Before importing the data, I cleaned up the date column a bit since it exported as a raw format as YYYYMM, so I formatted it as YYYY-MM.

A graph with red dots

Description automatically generated A graph with blue dots

Description automatically generated

After plotting the data, I can immediately see a clear upward trend, which suggests that there is an increase in temperatures over time. You can see that from the 1970’s on, ocean temperatures have consistently been higher than the average, which we know of as global warming. Between 1970 and 2004, greenhouse gas and CO2 emissions increased by 70% and 80%, respectively[[7]](#footnote-7). Additionally, I see there is a lot of variability in the data, which might be due to natural or man-made climate changes. The ocean specifically, aside from global warming and other disasters caused by man, can experience natural global impacts like El Nino and La Nina, which impact the weather, wildlife, and the ecosystem in general. “Episodes of El Nino and La Nina occur every two to seven years, on average, but they do not occur on a regular schedule”[[8]](#footnote-8). These natural occurrences could impact MoM and YoY results, so it is something to keep in mind when looking at seasonality and trends.

Unlike my other datasets, after taking the log value, I did not notice much change in the plot. The trends, outliers, and variance all looked the same. Based on this, I do not see the need to use the log value data for my time series analysis. After plotting the lagged values, you can see there is an extremely high correlation in the lagged values. There are also minimal values that A graph of a scatter plot

Description automatically generatedare straying from the general clustering, which indicates that there are few outliers in the data (which tells me that the anomalies overall are consistent).

A number and a sign

Description automatically generated with medium confidence After plotting the data and observing the variance and mean, it looks like there might be an underlying trend that is causing the anomalies to increase over time. It looks like the spread of the data points is increasing over the years as well. This indicates that not only are the ocean anomalies rising but are also becoming more variable.

# **2: Threshold Models – Chapter 15 Analysis**

For part two of the final project, I decided to work on reviewing Chapter 15, which explores Threshold Models. I thought this would be an interesting chapter to explore as it might relate to some of my datasets.

## 2.1. Graphically Exploring Nonlinearity

Graphically exploring the data can help investigate whether your data is linear or nonlinear. The beginning of this chapter introduces ways to graphically identify if your data is nonlinear.

A red text on a white background

Description automatically generatedA group of graphs with dots

Description automatically generated with medium confidence In this chapter I learned that nonlinearity can be identified through various plots like scatter plots and regression lines. The way to do this is by using a nonparametric regression curve to model the lagged relationship in your time series data. The technique commonly used to create this nonparametric regression curve is with the Nadaraya-Watson estimator. This technique estimates the mean of a response variable based on its lagged values by averaging all nearby observations. The proximity and its weight are usually determined by the standard normal probability density function. This means that the Nadaraya-Watson estimator gives more weight to observations that are closer to the point of interest. For a more accurate estimate, polynomial estimators can be used to better help capture the relationship. These polynomials are fitted around the points of interest.

In the first example provided in the book, the authors use the ‘arima.sim’ function to simulate a time series based on an ARIMA(2,1) model. The ‘set.seed’ function creates a random number generation for the plots. Then, the ‘lagplot’ function creates the scatterplots of the time series against its own lags.

In Exhibit A, this example helps the reader understand what they should look for in these plots to identify nonlinearity in their data. If the data is normal and linear, the lag-1 regression plot would reflect that with a linear line, meaning that the nonparametric estimate line would be straight. If the line is not straight, this would indicate that the data is nonlinear. In the case of time series plotted in Exhibit A, the regression lines are relatively straight, which indicates that the process is in fact linear.

A white background with red text

Description automatically generatedA graph of a number of predators

Description automatically generatedUsing the Veilleux dataset, which observes the population fluctuation of a prey-predator system, I will look at the log transformed data of the predator values. The plot to the right shows two things: stationary data (visualized by a solid line) and the lower regime of a fitted threshold (indicated by the solid circles). Essentially what this plot is showing us is the times where the series has settled into a pattern and where the predator popular has settled into a lower regime.

A group of graphs with black lines

Description automatically generated with medium confidenceI will now observe the lagged regression plots for real data to identify whether the data is linear or nonlinear. After plotting the lags, I can clearly notice that the estimate regression functions are nonlinear, specifically for lag-2, lag-3, and lag-4, which indicates that the data overall is nonlinear. Additionally, the large holes in the center of the plots also support that the data is nonlinear.

To summarize, to identify whether data is linear or nonlinear with the use of graphics, you must first plot the data and build a nonparametric regression curve to help visually identify linearity in the data. The regression curves can help you graphically identify whether your data is linear or nonlinear.

A group of black and white graphs

Description automatically generatedBelow is the graphical test for nonlinearity with my Citi bike data. Visually it is hard for me to see the regression curve due to the large amount of data in the plot, however, I can interpret this as a strong positive relationship in the lag plot and assume there is a linear relationship. Additional testing, which will be discussed in the next section below, is needed to identify linearity in the data.

## 2.2. Tests for Nonlinearity

In addition to graphically observing whether the data is linear or nonlinear, some tests have been developed to assess whether there is a need for nonlinear modeling in timeseries analysis. The main tests discussed in this chapter are by Keenan’s and Tsay, which are both a form of a Langrange multiplier test. As a refresher, a Langrange Multiplier is a statistical test that evaluates the need of adding parameters to a model.

The first test that is detailed in this chapter is the Kennan Test. Keenan’s test for nonlinearity uses a second-order Volterra expansion to approximate nonlinear stationary time series. If linear, the double sum expansion vanishes, and if nonlinear, you must check whether the sum is significantly different from 0. Below is an outline of the steps needed to test for nonlinearity using Keenan’s test:

1. Perform a linear regression of Yt on it’s past values (Yt-1, Yt-m), with m being a prespecified positive integer, and then calculate the values, residuals, and the residual sum of squares.
2. Regress the squared residuals on the past values, making sure to include an intercept term, and then calculate the residuals from this regression.
3. Regress the original residuals without an intercept. Calculate Kenaan’s test statistic by multiplying (*n* - 2*m* -2) / (*n – m* - 1) to the F statistic from the last regression.

A graph of the sunspot numbers

Description automatically generatedUnfortunately, Keenan’s test is only good for detecting nonlinearity in the form of square of the mean. Because of this, Tsay extended Keenan’s test by considering a more general form, allowing for a broader set of nonlinear alternatives by replacing the square of the linear predictor with a more general set of terms. Tsay’s test checks if all the coefficients of the quadratic terms are zero, and if they are not, that suggests the presence of nonlinearity.

In the textbook, they example these tests with the Spot dataset (plot displayed on the right). In this dataset, we assume that m = 5 (which is based on the AIC). For Keenan’s test, the p value is 0.0002. For Tsay’s test, the p value is 0.0009. The results of these tests both indicate that the data is nonlinear. While these tests are useful in detecting if the data is nonlinear, to me it seems as though observing this graphically is not only easier but quicker to identify as well. While the TSA library does offer script to help perform these tests, I do think most people are visual learners and being able to quickly see nonlinearity might be a better fit for some.

A screenshot of a computer code

Description automatically generatedAfter graphically observing the predator dataset, it was clear to me that the data was nonlinear. To confirm this, I used the Keenan test function. The Keenan.test() function, which takes in three arguments, X- the time, order- the working AR order (and if this argument is missing it is estimated by minimizing AIC via the AR function), and the last input is a user-supplied option for the AR function. For the textbook data, I just used the function as it. It gave us a very low P value and a order of 2.

A screenshot of a computer code

Description automatically generatedHowever, for my Citi Bike data, I played around with different orders for the autoregressive model. After plotting the autocorrelation when I first was exploring the data, I noticed it was highly correlated. So, I decided to incrementally increase the order to see if specific cycles might have an influence on bike usage. As seen from the different tests on the right, the different order values do suggest that there is nonlinearity in the data. At order 3, the P values is extremely low, which suggests that there is strong evidence against linearity at this order.

For order 7, the P value is .001092284, which also gives evidence to nonlinearity in the data. Order 14 is when you can see the evidence decreasing, however, still suggests that the data is nonlinear. Finally, at order 30, the P value is just above the .05 threshold which indicates marginal significance.

Based on the results from the different orders across Keenan’s test, this tells me that the data is in fact nonlinear. This tells me that looking at just graphical or just a test or nonlinearity might not be enough to decide whether data is linear or nonlinear. The use of both is helpful in better understanding my data.

## 2.3. Polynomial Models

A graph with numbers and lines

Description automatically generatedThe book discusses the limitations of polynomial regression models, particularly as polynomial time series models with a degree higher than 1. These models are considered unsuitable due to their tendency to diverge into infinity, as this characteristic limits the use for prediction. In the simulated example, I can see that the time series becomes explosive when t = 15. This explosive characteristic is confirmed for any polynomial AR process of degree higher than 1.

Additionally, the book also states that while polynomial AR models with bound errors can have stationary distribution (examples provided below), making them useful can be difficult. The logistic plot below shows an example of how the data could exhibit stationary behavior. However, the bound of the noise distribution varies with the model parameters and the initial values, which complicates their use in time series[[9]](#footnote-9).

A graph with lines and numbers

Description automatically generatedA black lines on a white background

Description automatically generated

## 2.4. First Order Threshold ARIMA / Higher Order Threshold

This is where we are finally introduced to threshold models in nonlinear time series analysis. A threshold model, also known as piecewise model, can switch between different linear submodels based on the value of a threshold variable. This type of model is extremely helpful when representing time series that behaves differently under various conditions.

There are many variations of the threshold model, however, the text specifically focuses on the self-exciting threshold autoregressive model, commonly referred to as SETAR. A SETAR model switches between different linear submodels depending on whether the lagged value crosses a specific threshold. This switch is dictated by the time series itself, which is why the model is described as “self-exciting”.

Additionally, it is important to know that first-order TAR models can be generalized to a higher order and delay with varying autoregressive orders for each regime. While this can be done, stability gets more complicated the higher the order. However, these models can handle multiple regimes. This discussion in the text primarily focuses on the two-regime case.

A math equation with red text

Description automatically generatedThe first example the text provides is a time series plot of a simulated TAR model. Like some earlier examples in the text, I use the “set.seed” function to create a random number generator, the “tar.sim” function is used to simulate a first-order TAR process by setting the number of observations to 100, setting the autoregressive parameter for the first A graph showing a number of numbers

Description automatically generated with medium confidenceregime and second regime, and some additional inputs to build the simulated model. Then the data is plotted so I can observe the series graphically. This simulated tar model gives me insight into the cyclical behaviors with asymmetric cycles which indicates inversibility. This is something we tend to see in some real-world nonlinear time series.

The authors also create a Q-Q plot, which shows that the simulated data has a thicker tail, indicating nonnormality. Additionally, here are the Q-Q plots for the chapter predator data along with my Citi Bike data.

A graph of a graph

Description automatically generatedBelow are the Q-Q plots for the predator and Citi Bike data. As I move through the sections of the chapter, I am attempting to follow the steps the authors are using to understand the data. The Q-Q plot of the predator data indicates deviations in both tails, and the data points don’t really follow the reference line. The Citi Bike data indicates that while some of the plot points follow the reference line closely, you can see deviations at both tail ends. This mimics what we saw with the sample data from above.

A graph showing a line

Description automatically generated with medium confidenceA graph of a number of dots

Description automatically generated with medium confidence

## 2.6. Testing for Threshold Nonlinearity

Earlier I learned that we could use tests to detect nonlinearity in our data, however, the Keenan and Tsay tests are used for quadratic linearity. These tests are not appropriate for threshold linearity. To test for this, we need to use a ratio test. In a ratio test, the null hypothesis assumes a linear autoregressive model against an alternative of a two-regime threshold autoregressive model, with equal noise variance across both regimes.

A screenshot of a computer code

Description automatically generated The textbook provides the code for this test. The first line, pvaluem=NUL, is where we will store the test results. We then set up a function that loops through how many delays we set up. We then enter in our parameters, with ‘p=5’ being the autoregressive order parameter, ‘d=d’, being the delay value that is being tested in the loop, ‘a=.25= and ‘b=.75’ being the percentiles. After that, the rest of the code combines the results and then prints them out so we can read the results for each delay with its test statistic and p value. For the spot dataset, all the p values are less than 0.000, and indicates nonlinearity.

For the Citi Bike data, the results suggest that there is significant threshold nonlinearity in the Citi Bike data. As you can see at delays 4,5,6 and 7, they all have very small p values. This tells me that the Citi Bike data is nonlinear and might warrant a threshold model. I do know that my data is highly seasonal, so understanding whether the model would be better with a threshold model or seasonal ARIMA is something I’ll explore later.

A screenshot of a computer

Description automatically generated

## 2.7. Estimation of a TAR Model

Moving into estimating a TAR model with the predator data. After going through this section, it took extra effort to understand and interpret the TAR model and results. I have come to learn that a lot goes into these models, and while I am still not 100% positive in my understanding of TAR models, I believe I have come to understand most of it. For this section I am only focusing on building a TAR model with the predator data provided by the book as I struggled with my own data, which I believe is due to the seasonality. You will also later find I built a reasonable ARIMA model for my Citi Bike data and a TAR model was not necessary.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer code

Description automatically generated Below is the code to the estimation methods with the predator data. We set the maximum order to 4 (determined by the AIC). Below displays the nominal AIC values, where the smallest value is when d = 3, leaving us to estimate the delay to be 3. This method selects the order 1 for the lowest regime and the order 4 for the upper regime.

We then plot the t statistic and p values to better understand the upper and lower regimes. We can see that the lower regimes correspond to “increasing phase of predator cycles, and the upper regime corresponds to the decreasing phase of the predator cycles”[[10]](#footnote-10). This visualization helps address challenges of estimating the parameters in the TAR model.

Initially the text explores a TAR (2;1,4) model. This is chosen to help capture the dynamics of the system with more complexity. The plot below helps us understand the cycles. As you can see, this does its job at capturing the cyclical nature of the predator data. You can also get an idea of the switching behavior between the different regimes. This plot of the TAR(2;1,4) model is stationary.

**TAR (2;1,4):**

A graph of a graph

Description automatically generated with medium confidenceA graph of a graph

Description automatically generated

The text then creates a more simplified model of TAR(2;1,1). This was created to help examine a more parsimonious model that reduces complexity while retaining the important aspects of the regime switching. This was done by constraining the maximum order of 1. From the plot below you can see that the model is a simpler representation of the behavior which in turn makes it easier to interpret. In summary, while the TAR(2;1,4) model helps better capture cyclical behavior, the TAR(2;1,1) better captures asymmetry.

**TAR (2;1,1):**

A line graph with dots

Description automatically generated with medium confidenceA graph of a graph

Description automatically generated

## 2.8. Model Diagnostics

Looking at the model diagnostics for TAR(2;1,1), I can see that the plot of the standardized residuals shows no significant pattern except for one possible outlier. The ACF plot suggests that lag 1 of the residual autocorrelation is significant which indicates a lack of fit for the model. For TAR(2;1,4) we see that the residuals also show no pattern, the ACF shows no significant autocorrelation. The Q-Q plot also shows that the standardized residuals also appear relatively straight. Based on these results, we can conclude that the TAR(2;1,4) model provides a good fit for the predator data.

**TAR(2;1,1) TAR(2;4,1) Q-Q Plot (2;1,4)**

A graph with numbers and lines

Description automatically generated A graph of numbers and lines

Description automatically generated with medium confidenceA graph of a normal qq plot

Description automatically generated

## 2.9. Prediction

Lastly the textbook delves into the problems of predictions from a TAR model. Unlike ARIMA models, TAR models’ prediction distributions are typically nonnormal, so prediction intervals might need to be computed using simulation techniques. For the predator TAR(2;4,1) model with a parameter value of d=3, the visualization below gives us the prediction intervals. The outer lines represent the 2.5th and 97.5th percentiles of the predictive distribution and the middle line represents the median. While the patterns initially follow the predator data, the further out the predictions go into the future, the more they approach a median value.

A graph of a graph

Description automatically generatedA graph with lines and numbers

Description automatically generated

To conclude this chapter, we learn that TAR models are useful when it comes to predictions with data that exhibits nonlinear behavior. The prediction intervals allow you to capture uncertainty in the data while also capturing cyclical behaviors. However, these models are difficult to build and understanding the thresholds for the different regimes can take some time and patience to figure out. While I started to work through my Citi Bike dataset with this chapter, I learned that the data (despite passing the tests for nonlinearity) was not appropriate for this chapter due to the heavy seasonality. While I believe to have a grasp on threshold models, this by far was the hardest chapter for me to work through in the textbook, not only due to the model selection but all the tests to determine if the data would fit a TAR model.

# **3: ARIMA Models**

I will start modeling with my KSS and WDC stocks. For stock data - depending on what the goal of the forecast is depends on the approach. If the focus is to forecast future returns, then I would stick to an ARIMA model. If I was interested in the risk, I would move forward with a GARCH model. In practice people do use a combination of both models, called an ARIMA-GARCH model. This can be used for better stock analysis and trading strategy development[[11]](#footnote-11). For this project, I will first fit an ARIMA model, and if the residuals do not have any autocorrelation, I will move forward with the model. If not, I will attempt to also fit a GARCH model on the residuals to better help capture volatility. I will use this method on both of my stock portfolios.

## 3.A KSS Model

The first time series model I will be building is for Kohl’s Corporation (KSS). Daily returns are fundamental in financial time series analysis but are tricky to work with. As mentioned earlier, I am going to fit the data with an ARIMA model, analyze the residuals, and then validate the model.

The ACF plot below shows that most autocorrelations are within the bound lines, however a few are outside the bound lines (around lag 8, 18, 26, 27, and 31). PACF looks like ACF as there is some partial autocorrelation in some of the lags.

A graph with lines and numbers

Description automatically generated A graph with lines and numbers

Description automatically generated

After fitting my KSS data with ARIMA(1,0,0), I get the below results for my residuals. The residual plot shows some randomness, however, there are some large spikes in the beginning and end of the plot which do suggest some large errors. The ACF plot shows autocorrelation at various lags which means the model is not capturing autocorrelation. The histogram shows heavy tails. These findings, coupled with a low p-value from the Ljung-Box test tell me the that this ARIMA model is not a goot fit. Due to these results, I will move forward with fitting a GARCH model on the residuals. Unfortunately, using the garch function in TSA led me to some difficulties and error in my outputs. Fortunately, I was able to find some other libraries in R that also perform GARCH models. I will be using the fGarch[[12]](#footnote-12) library as it allows for easy garch forecasting.

A screenshot of a graph

Description automatically generated

A white background with black text

Description automatically generated

A graph with lines and numbers

Description automatically generatedAfter testing a few different models, the best outcome with the lowest AIC was GARCH(1,1). This model showed that the plotted standardized residuals were random and did not show any patterns or trends, the ACF of squared standardized residuals were all within the confidence bounds, and the Ljung-Box tests on the standardized residuals suggests that the model captured serial correlation and volatility clustering. However, Q-Q plot indicated that the residuals have heavier tails than a normal distribution. Fortunately, ‘fat tails’ are common in financial data and are not inherently negative. These tails reflect the reality of market behavior and risk, and by acknowledging that they exist, financial models can be adjusted to better accommodate risk assessment and decision making[[13]](#footnote-13).

A screenshot of a computer

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A line graph with a red line

Description automatically generated

Next, I will forecast the ARIMA model and forecast the volatility from the GARCH model. I will combine the forecasts and then plot the data. I will look at a 100-day forecast.

A graph with lines and text

Description automatically generated with medium confidence

For the most part, the GARCH model appears to be capturing most of the potential values within the GARCH Volatility Bands. While only 6% of actual returns fall outside of the bands, while the remaining 94% fall within those bands. To me, this looks like the model is doing a pretty good job at capturing the volatility of the returns. While the model does not do a great job at forecasting returns, this could be due to it not finding any clear trend in the data.

## 3.B WDC Model

I will follow the same steps for WDC as I did for KSS, starting with the ACF & PACF plots. The ACF and PACF for WDC look similar, with both showing some lags within the bounded lines and some outside, indicating autocorrelation in the lags. I will perform an ARIMA model on the data, and then analyze the residuals to determine if a GARCH model is necessary. A graph with lines and numbers

Description automatically generatedA graph with lines and numbers

Description automatically generated

After fitting my model, I got an ARIMA(0,0,1). I can see that there is some randomness in the residuals, but in the beginning and end of the series there is some increased volatility. The ACF plot shows that there are many lags outside ther bounded lines which indicate there is still autocorrelation the model did not capture. The Ljung-Box test and histogram (which shows long tails in the plot) of the residuals indicate that an ARIMA is not a good fit and there is still autocorrelation in the lags. Based on these results I will need to fit a GARCH model.

*A screenshot of a graph

Description automatically generated*

*A white background with blue text

Description automatically generated*

For WDC, I used the GARCH(1,1) model. After fitting the model and viewing the residuals, I can see that the plotted residuals look random, the ACF plots shows that all but one lag fall within the bounded lines, meaning it captured autocorrelation in the model. The Ljung-Box test also indicates that the model has captured correlation. Like KSS, the Q-Q plot does show fat tails, but as I learned earlier this is common in financial time series. Overall, this model looks like it does a good job at capturing volatility in WDC returns.

*A graph with lines and numbers

Description automatically generatedA graph showing a graph of a wave

Description automatically generated with medium confidenceA graph of a line

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated*

Below is the WDC plot of actual returns, ARIMA forecast, and the GARCH volatility bands. For this model, it looks like volatility has been captured very well. All but one return is within the banded lines. This tells me that the model is a good fit. Unfortunately, I have the same issue with my forecasted values as I did with my KSS model. It looks like the model does not do a good job at forecasting, which means the model does not do a good job at capturing any trends in the data.

A graph of a graph

Description automatically generated

For this model, I would recommend WDC use it for risk management. From this 100-day forecast, it’s easy to see that the model does a good job of capturing volatility. For stakeholders at Western Digital Corporations, this model could be useful for estimating future volatility, volatility targeting, stop loss rules, and many more analysis that require these volatility bounds.

For both KSS and WDC, it looks like the GARCH models are capturing most of the potential values within the volatility bands. While some refinement could be done to improve the ARIMA forecast of the values, I am confident with the volatility performance. While I would not recommend these models for forecasting future returns, I would recommend this model for risk assessment. These models could be used for portfolio management or option pricing. If Western Digital Corporation or Kohl’s wanted to better forecast returns, they might need to consider including additional features or building a more complex ARIMA model. However, despite acceptable performance, I would still suggest future refinement on the data to make sure that the models are able to capture volatility for longer future forecasts, especially during time of extreme variance.

## 3.C Citi Bike Model

A graph of lines with text

Description automatically generated with medium confidenceFor my Citi Bike model, I have decided to incorporate exogenous regressors (average temperature and total rainfall). I am hoping that they can help me build a more reliable model. Since the goal is to forecast demand as accurately as possible, I believe adding in weather data will be useful as the weather will impact someone going outside to ride a bike. I also broke out my training data to include from January 2018 through June 2023. My test data will be from July 2023 through October 2023.

A graph with lines and numbers

Description automatically generatedBelow are the ACF and PACF plots for Citi Bike daily bike count. The ACF plot shows a slow decay in the lags, which is indicative of non-stationary time series. I also notice that there appears to be some level of periodicity in the data, shown by the dips and peaks in the lags. This tells me that that there is some level of seasonality in the dataset, which was also confirmed by viewing the time series plot in part 1. When looking at the PACF plot, I can see there is a significant drop off from lag 1 to lag 2. Up until lag 7 I can see there are several spikes above the bound lines, and after lag 7, the lags remain within the bounded lines for the most part. This tells me after lag 7, those lags do not have significant autocorrelation and might not need to be included.

Below is a plot to show seasonal decomposition in the data. The first panel at the top, labeled ‘data’, just shows the plotted data over time. The second panel, called ‘seasonal’, shows the seasonal component that has been extracted from the data. Based on this panel, I can see there are some clear and consistent cycles in the data. There are obvious dips and peaks throughout the time series. The third panel, labeled ‘trend’, shows the long-term movement in the data after removing the seasonal effects. In the Citi Bike data, there is a very clear upward trend in the data (despite the little drop off near the middle). As mentioned earlier, this might be due to a rise in bike demand in NYC. Lastly, I look at the bottom panel labeled ‘remainder’. This is the residual component after seasonal and trend components are removed. What this should essentially represent is white noise. I do notice there is a pattern and some consistent spikes in the data. This could represent some seasonal or trend components that were not captured by the data. Based on this plot below, the decomposition shows that there are significant seasonal patterns in my data, as well as a strong long-term trend. Because of these strong seasonal trends, I am going to assume that I might need to use a Seasonal ARIMA model.

A graph of a graph

Description automatically generated with medium confidence

Based on the ACF plot, this tells me that the data needs differencing (d=1). The significant spike at lag 1 indicates that the model needs only a few AR terms, so I will start with p = 1. Looking at the seasonal decomposition plot, there are strong seasonal trends in the data which tells me that the seasonal MA terms also need differencing. I will move forward incorporating these into the model.

As mentioned earlier, I decided I will be incorporating average temperature and total rainfall as exogenous regressors into the model. With the above insights in mind, I experimented with values for p ranging from 0-2. Based on the results below, the ARIMA(1,1,1)(0,1,1) was the best fit based on AIC and BIC. My next step will be to test the model on known outcomes from the dataset.

A screenshot of a computer code

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Model (period=7) | AIC | BIC |
| Arima(1,1,1)(1,1,1) | 364.05 | 43.25 |
| Arima(2,1,1)(1,1,1) | 365.95 | 410.75 |
| Arima(1,1,1)(0,1,1) | 362.87 | 396.46 |
| Arima(0,1,1)(1,1,1) | 409.49 | 443.09 |
| Arima(0,1,1)(0,1,1) | 407.53 | 435.53 |

After plotting the residuals, ACF of the residuals, and performing the Ljung-Box test, I can see that this model is in fact a good choice. The plot of the residuals is random and does not indicate any patterns, despite being some spikes near the middle of the plot. The ACF of the residuals are all within the bounded lines (for the most part), and the Ljung-Box test (p-value of .908) indicates that there is no significant level of autocorrelation in the residuals. All these tests leave me to believe that this model is a good fit.

A graph of a bicycle model

Description automatically generated A graph with lines and text

Description automatically generated with medium confidence

A white background with red text

Description automatically generated

To assess the predictions, I forecasted the next 4 months of data and validated the results on the test dataset. Below you will find the plotted forecast, the actual values, and the forecast accuracy. Overall, the model appears to be performing reasonably well. The errors are within an acceptable range and the residuals are not autocorrelated. While the MAPE test set is higher than the training set, this is expected behavior as out-of-sample predictions tend to have higher error rates.

A graph of a graph

Description automatically generated with medium confidence

A graph showing a graph

Description automatically generated with medium confidence

A number with numbers on it

Description automatically generated with medium confidence

After validating the model on out-of-sample data, I am confident with the performance of the model in its ability to forecast bike ridership. Although the MAPE is somewhat elevated, the model still holds considerable potential to inform Citi Bike in strategic decision making. From an operational standpoint, it offers valuable insight for optimizing inventory allocation and workforce planning, which in turn enhances cost-efficiency and mitigates financial risks associated with demand variability. From a marketing/advertising perspective, ridership predictions can be used to capitalize on increased usage and encourage users to sign up for memberships, or even offer promotional deals during slower seasons. There are countless other business use cases for this model, however, if accurate ridership data is desired then some additional refinement is needed.

To further refine the model’s predictive accuracy, I would advise including additional variables. The current model already benefits from integrating average temperature and rainfall data, so incorporating additional variables like holiday schedules, special events, public transit utilization, and competitive transport options could substantially elevate its forecasting precision.

## 3.D Ocean Anomaly Model

Next, I will build my Ocean Anomaly model. From the below ACF plot, I can see that the lags show a slow decay which indicates that the data might not be stationary and will need differencing. For the PACF plot, I can see that lag 1 shows significant autocorrelation and is marked above the bound lines. However, all lags after lag 1 fall within the bounded lines and suggest they are not significantly correlated.

After plotting the seasonal decomposition, I can see clear seasonal patterns in the data. The seasonal panel indicates a strong cyclical pattern. This makes sense since ocean data tends to have strong natural climate cycles. The trend panel indicates a slight upward trend with a lot of fluctuations. Lastly, the bottom remainder panel shows random fluctuations in the residuals which do not indicate clear patterns, informing me that the model has captured most of the systemic information in the data.

Based on this information, the sharp drop in the PACF after lag 1 suggests that an AR(1) model might be appropriate, and due to the seasonal pattern in the decomposition a seasonal AR or MA component is needed.

A graph with lines in the middle

Description automatically generatedA white paper with a line

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated

Like the Citi Bike model, I experimented with different variations to determine the lowest AIC and BIC values. Unfortunately, a lot of the models I tested did not do a great job with forecasting. I even tried using the auto.arima function to see what it recommended, which ended up being an Arima(2,1,3) with no seasonal components. I tested it out and it ended up performing horribly, so I knew it was best to continue with a seasonal model. I discovered that the auto.arima function did not give me the best outcome for my Citi Bike data either and I had better results looking for the best fit on my own. For the ocean anomaly data, the best fit ended up being Arima(2,1,3)(1,1,1)[12]. I decided to move forward with fitting this model and plotting the residual analysis.

A screenshot of a computer

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Model (period=12) | AIC | BIC |
| Arima(1,1,1)(1,1,1) | -4588.46 | -4562.49 |
| Arima(0,1,0)(1,1,1) | -4590.02 | -4574.44 |
| Arima(2,1,1)(1,1,0) | -4129.14 | -4103.17 |
| Arima(3,1,0)(0,1,0) | -3686.43 | -3665.65 |
| Arima(2,1,3)(1,1,1) | -4621.24 | -4579.69 |
| Arima(3,1,3)(1,1,1) | -4619.33 | -4572.59 |

The plotted residuals over time do not show any apparent patterns of trends. They all seem random and fluctuate around zero. The ACF of the residuals show that all lags (except for one) are within the confidence bounds and implies that there is no autocorrelation in the residuals. The Ljung-Box test gave me a p-value of 0.9963, which suggests that the residuals resemble white noise. Lastly, the histogram of residuals appears to be symmetrical and suggests that the residuals are normally distributed. All these tests indicate that the residuals do not exhibit any autocorrelation and do not show any seasonality or trends which indicate that this could be a good model fit. Next step is to validate with out of sample values.

A graph showing a wave of ocean animals

Description automatically generated with medium confidenceA graph with lines and numbers

Description automatically generated with medium confidence

A white background with red text

Description automatically generatedA graph of a normal distribution with Ryugyong Hotel in the background

Description automatically generated

A close-up of numbers

Description automatically generated Below I have plotted the forecasted values along the actual values. While the model seems to fit the training data well (low error and no autocorrelation in the residuals), the performance on the test set did not do well. There were high error measures across the board, and the model forecasts deviated significantly from the actual values in the test set. This is also visible by looking at the plot of the forecasted vs. actual values. You can see the model attempting to identify trends and patterns; however, the nuance of ocean anomalies was not captured.

A table with numbers and text

Description automatically generated

A graph showing the weather

Description automatically generatedAfter evaluating the forecasts, I would not be confident to recommend this model to a client. Ocean temperatures are influenced by many factors-not just time. Other environmental factors like solar cycles, volcanic activity, carbon dioxide levels, etc. play a role in the changes in ocean temperatures. I think this model would benefit greatly from additional external predictors to help with forecasting ocean anomalies. A study from the Journal of Climate discusses the use of a suite of 13 coupled global atmosphere-ocean models for the seasonal prediction of sea surface temperature anomalies[[14]](#footnote-14). This is just one example of how many features might be needed to make an accurate prediction in climate data. I believe a more sophisticated model is needed to accurately forecast ocean anomaly data. My recommendation would be to include as many environmental factors as possible to help refine and improve forecast accuracy.

# **3: Conclusion**

Throughout this semester dedicated to learning about time series modeling, I've come to realize that mastering this discipline might span years, or even longer. While I've successfully implemented models that yield reasonable outputs, perfecting time series analysis, especially within financial data and stock market predictions, remains an elusive goal.

Despite not achieving the desired goal of being able to forecast ocean anomalies, I did learn that environmental data is one of the hardest problems to tackle. Through my research I learned that scientists utilize many different features and models to forecast accurately, and a lot of those resources are not at my disposal. However, even though my model did not accurately forecast ocean anomalies, it did highlight the rise in ocean temperatures which we know exists in today’s world.

My experience with building the Citi Bike model is my highlight of this project. Despite one of my learnings being that I might need some further refinement for more accurate forecasts, I felt as though I was able to create a successful time series model. My goal after this class ends is to continue developing this model and improve its performance as I would love to add this model to my machine learning portfolio.

Overall, this class was challenging yet rewarding, and it has shown me that this type of analysis will require continuous learning. However, I am optimistic about the future refinement for my Citi Bike model. I hope to continue to deepen my understanding of time series analysis as I implement my skills from this class into uses cases at my job or into future work.

1. LN3.A Foundations of Time Series Analysis, Page 16. [↑](#footnote-ref-1)
2. LN3.A Foundations of Time Series Analysis, Page 20. [↑](#footnote-ref-2)
3. Downey, L. (2023, April 24). Efficient Market Hypothesis (EMH). Investopedia. https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp [↑](#footnote-ref-3)
4. <https://mesonet.agron.iastate.edu/request/download.phtml?network=NY_ASOS> [↑](#footnote-ref-4)
5. Wang, H., & Noland, R. B. (n.d.). Bikeshare and subway ridership changes during the COVID-19 pandemic in New York City. Edward J. Bloustein School of Planning & Public Policy. Retrieved December 15, 2023, from <https://bloustein.rutgers.edu/bikeshare-subway-ridership-changes-during-covid19-pandemic-nyc/> [↑](#footnote-ref-5)
6. Farías, A., & Harris, D . (2022, February 24). Opinion: NYC Must Commit Public Funding to Expand Citi Bike. City Limits. Retrieved December 15, 2023, from <https://citylimits.org/2022/02/24/opinion-nyc-must-commit-public-funding-to-expand-citi-bike/> [↑](#footnote-ref-6)
7. National Geographic Society. (n.d.). The Greenhouse Effect and our Planet. National Geographic. Retrieved December 15, 2023, from <https://education.nationalgeographic.org/resource/greenhouse-effect-our-planet/> [↑](#footnote-ref-7)
8. National Oceanic and Atmospheric Administration. (n.d.). El Niño and La Niña: Frequently asked questions. NOAA's National Ocean Service. Retrieved [2023], from <https://oceanservice.noaa.gov/facts/ninonina.html> [↑](#footnote-ref-8)
9. Cryer, J. D., & Chan, K.-S. (2008). Time Series Analysis With Applications in R (2nd ed.). Springer. [↑](#footnote-ref-9)
10. Cryer, J. D., & Chan, K.-S. (2008). Time Series Analysis With Applications in R (2nd ed.). Springer. [↑](#footnote-ref-10)
11. QuantStart. (n.d.). ARIMA+GARCH Trading Strategy on the S&P500 Stock Market Index Using R. Retrieved from <https://www.quantstart.com/articles/ARIMA-GARCH-Trading-Strategy-on-the-SP500-Stock-Market-Index-Using-R/> [↑](#footnote-ref-11)
12. <https://www.rdocumentation.org/packages/fGarch/versions/4031.90> [↑](#footnote-ref-12)
13. MacroSynergy. (2018, July 28). The dangerous disregard for fat tails in quantitative finance. Retrieved December 6, 2023, from <https://research.macrosynergy.com/the-dangerous-disregard-of-fat-tails-in-quantitative-finance/> [↑](#footnote-ref-13)
14. Krishnamurti, T. N., Chakraborty, A., Krishnamurti, R., Dewar, W. K., & Clayson, C. A. (2006). Seasonal Prediction of Sea Surface Temperature Anomalies Using a Suite of 13 Coupled Atmosphere–Ocean Models. *Journal of Climate*, 19(23), 6069–6088. American Meteorological Society. [↑](#footnote-ref-14)