**D213 Performance Assessment Task 1**

**TIME SERIES MODELING FOR MEDICAL DATA**

Fahim A. Akbar Student ID 001434895 Masters Data Analytics (January 1, 2021) Program Mentor: Mandy Rasmuson (385) 428-5957x5957 [fakbar3@wgu.edu](mailto:fakbar3@wgu.edu)

**Part I: Research Question**

**A. Data Mining Report Justification**

**A1. Research Question:**

The research question that this analysis will focus on is if time series modeling can accurately predict daily revenues for the hospital compared to the actual observed daily revenues.

**A2. Goal**

The goal of the analysis is to predict daily revenues based on the provided hospital data. To do so, 80% of the dataset will be used as a training set to predict the remaining 20%. This approach will generate values as a point of comparison, and help to avoid creating future predictions that cannot be evaluated using the given dataset. This project will be conducted using an ARIMA time series model to assess the 80% of data. Our aim is to accurately project the values for the remaining 20% of data.

**Part II: Method Justification**

**B. Assumptions of Time Series Model**

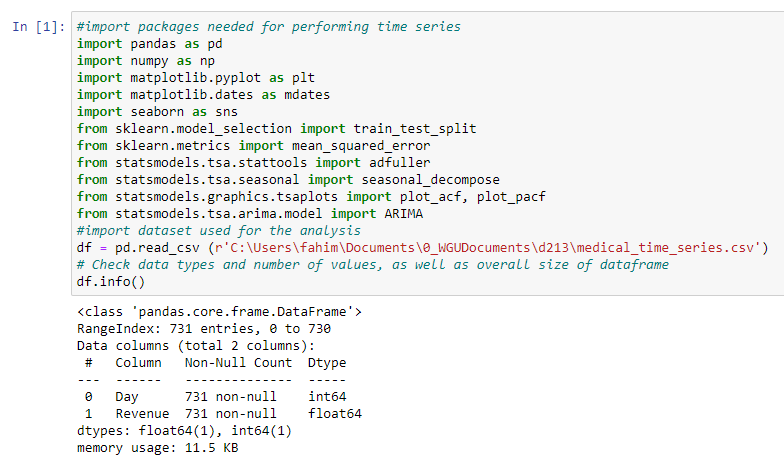
For this project, we will be conducting a time series analysis. Time series analysis using ARMA or ARIMA models operates on the following assumptions of the data being analyzed:

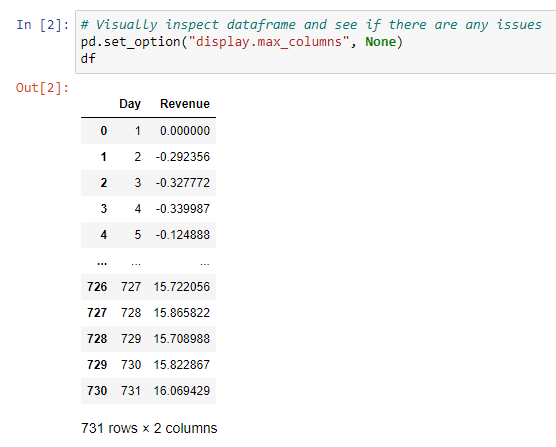
* The data used in the times series analysis must be stationary, and no trends or seasonality should be present. The mean and variance are stable when centered at zero after detrending the data, highlighting their importance in analyzing lags and differences as they represent specific points in time.
* There are no outliers or any other abnormal values present in the dataset.
* The data is univariate, and represents a singular variable intended for modeling
* Past datapoints are indicative of the behavior of future datapoints

Include the time series flaw shit in the conclusion/analysis

**Part III: Data Preparation**

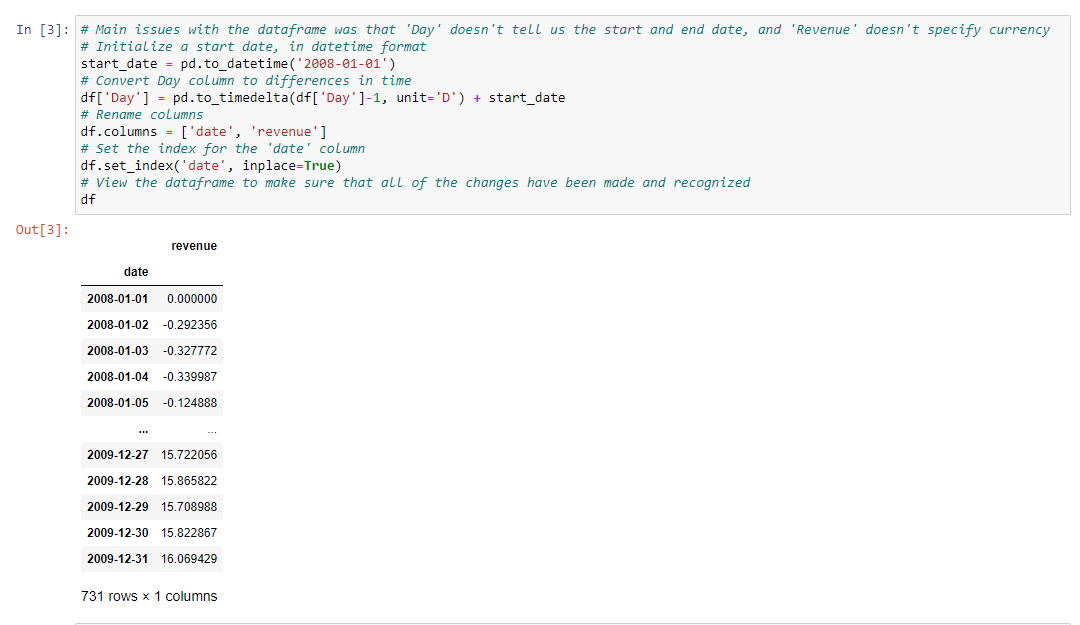
**C. Summary of the data cleaning process**

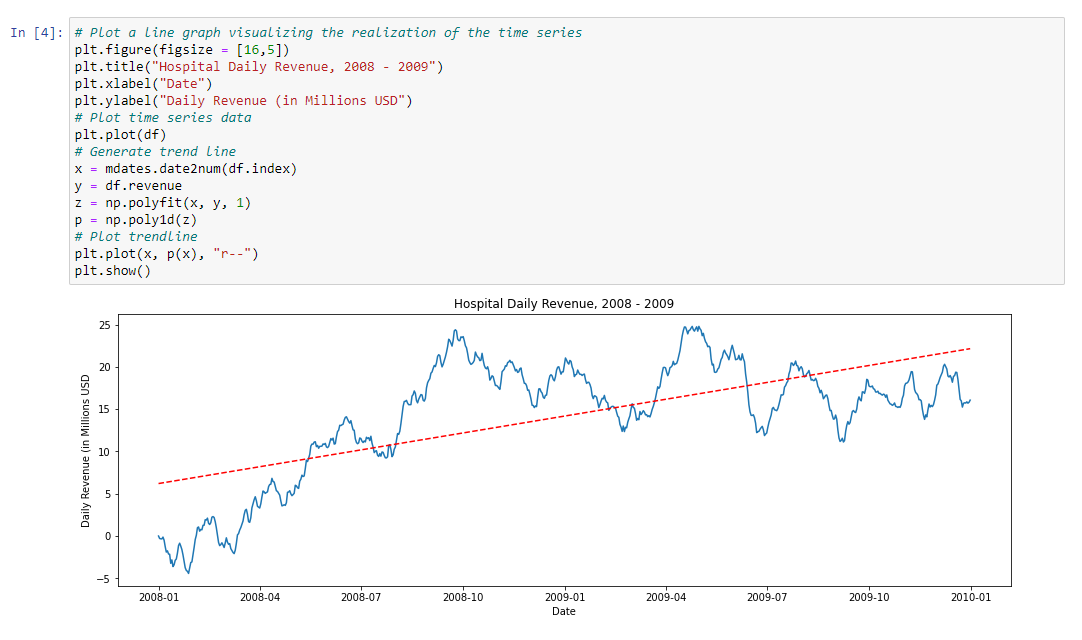
**C1. Time Series Visualization**

****

The two column dataset used in this analysis provides us with records of the first two years of revenue from the hospital. The first column, 'Day,' spans from Day 1 of the initial two-year period to Day 731. The second column, 'Revenue,' represents the daily financial performance for the respective day, indicating the hospitals monetary gains or losses in millions.

One limitation of the dataset present is the nature of the day column. The 'Day' column should be converted to a DateTime object for better usability, but the dataset lacks context about when the data was collected. It could represent two years starting from a random date, not necessarily recent years, and includes a leap year, resulting in 731 daily observations instead of the expected 730 for a two-year period. Since the start date for the data is never disclosed in the data dictionary as well, we will be using January 1 2008 as our starting point. Furthermore, it’s not specified what currency is used for revenue, so we will assume USD for this analysis.





After the data was cleaned and labeled, the graph provided above was generated to illustrate the daily revenue in millions of dollars USD for the hospital over the course of two-years, from2008 - 2009. The graph illustrates an increase in daily revenue over these years, which is further emphasized by the red trendline.

**C2. Time Step Formatting**

The time series is composed of a DateTime index and daily revenue observations in millions of dollars USD for the hospital. Each day, the DateTime index advances by one day, spanning from January 1, 2008, to December 31, 2009, with no missing data points. There are 731 data points in the sequence, including the leap year in 2008. Representing the data as a DateTime object allows for diverse aggregations such as revenue by week, month, or year, as well as other data manipulations.

**C3. Series Stationarity**

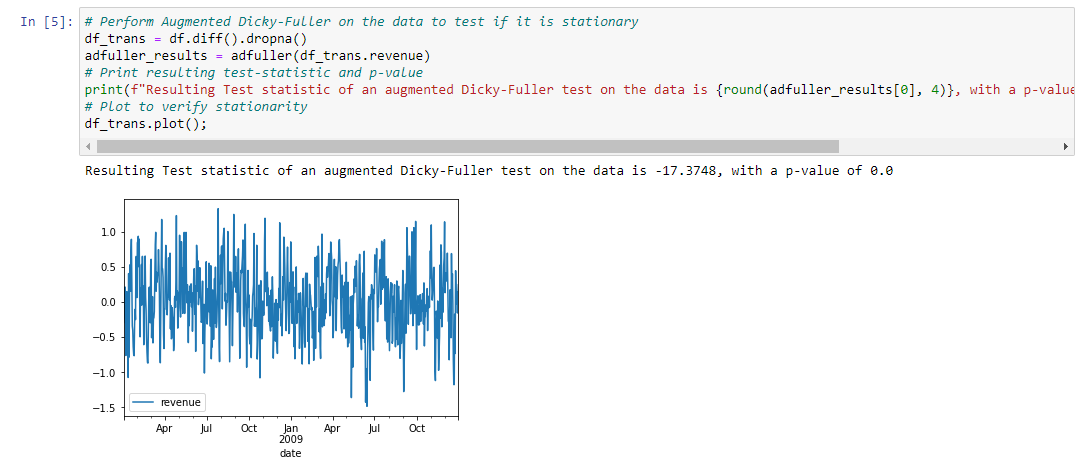
Stationary time series are characterized by the absence of trends. Therefore, the data here demonstrates a lack of stationarity.

In this dataset, the daily revenue of the hospital system initially hovered around 5 million dollars for the first few months of the analyzed period, eventually stabilizing between 15-20 million dollars per day.

This unmistakable pattern signifies a clear upward trend in daily revenue, and to confirm this trend, a trendline was included in the plot. The dashed red trendline clearly ascends, underscoring a distinct upward trend, leaving no room for ambiguity. Thus, it is evident that this data is far from being stationary.

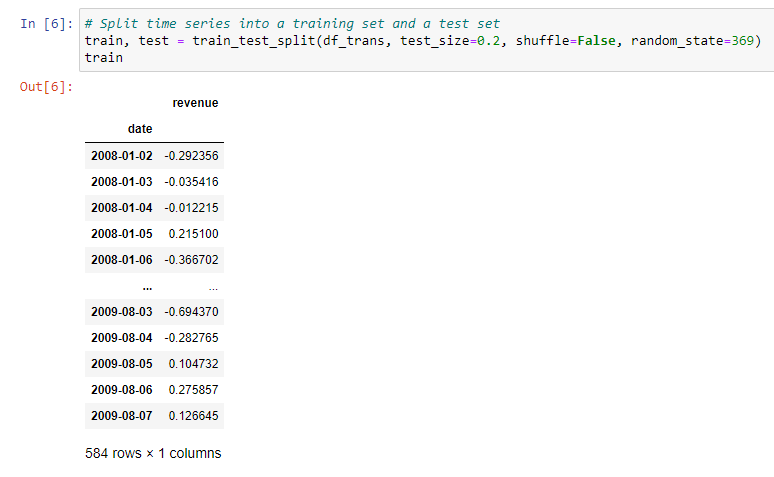
**C4. Data Preparation and Explanation**  
To prepare this data for time series analysis, several essential steps must be undertaken:

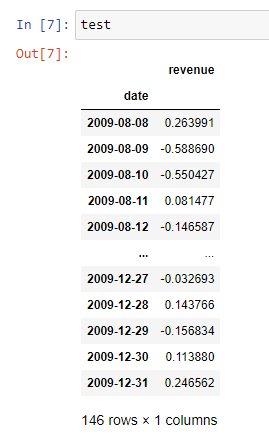
1. Convert the "Day" column into a valid DateTime format (refer to C1 above).
2. Set the revised "Day" column as the index for the time series (as explained in C1 above).
3. Transform the "Revenue" data into a stationary time series, removing any trend or seasonality.
4. Divide the data into distinct training and test sets.

****The plot and augmented Dicky-Fuller results shown above confirm that the data is now stationary, and we can continue with splitting the data into training and test sets. To ensure that both the train and test sets are provided and the time series retains its order, SciKit-Learn's train\_test\_split will be used with shuffle=False. This will retain the time series data’s order and prevent data rearrangement.

**C5: Copy of Prepared Data Set**

The data was successfully split into both a training and a test set. The training set comprises of 80% of the data and the test set comprises of 20% of the data. The training set is saved as 'D213Task1\_train\_clean.csv', while the test set is saved as 'D213Task1\_test\_clean.csv'.

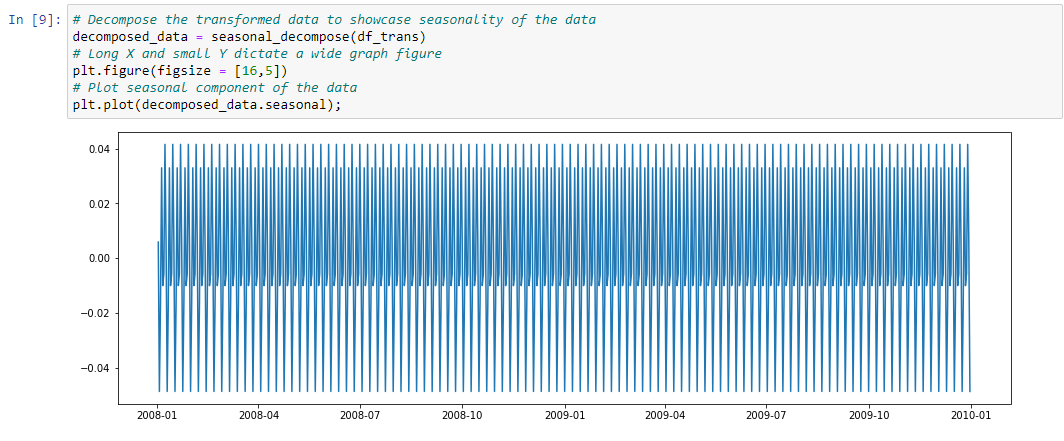




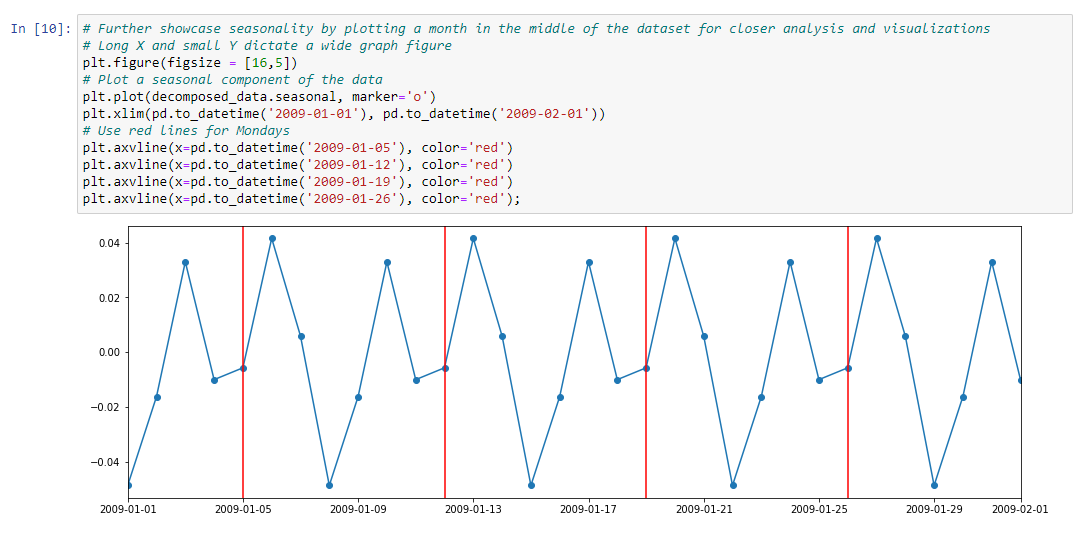


The preparation of the dataset was performed in a Python using a Jupyter notebook environment. The Jupyter notebook file is attached to the task submission. A pdf copy of the notebook and a txt. file of code used is provided with the task submission as well. A copy of the cleaned dataset is also provided with the task submission. Lastly, the entire code used is also provided at the end of the document.

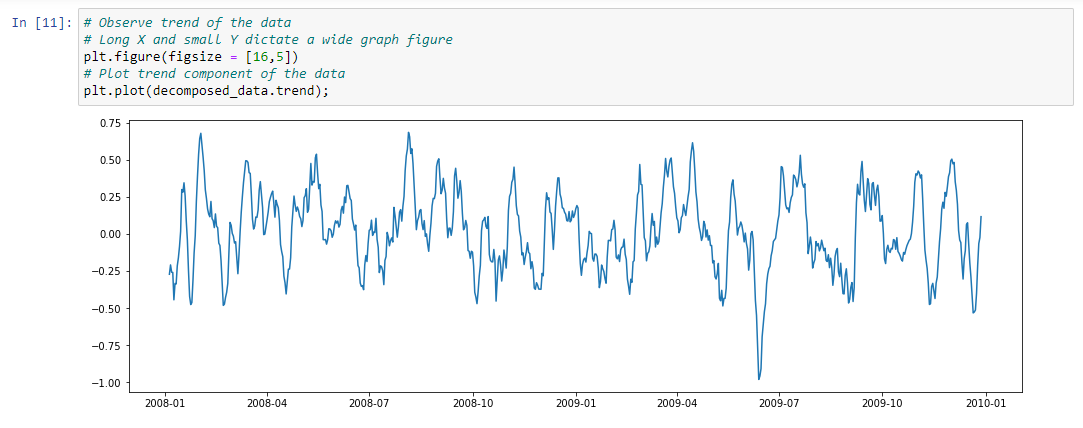
**D1: Annotated Findings & Visualizations**

Seasonality

By decomposing the data and looking at the seasonal component, we find a consistent, repetitive pattern. This indicates seasonality, though the magnitude of this seasonal component (as indicated by the y-axis) is very small, around 0.04 in either direction of zero at its most extreme, compared to an overall magnitude in the dataset of 1.0 - 1.5. Given the scope the observed seasonality (happening several times within a period of months), this is likely a phenomenon related to the days of the week. We can verify this, plotting a month in the middle of the dataset to take a closer look at the data.



In the above graph, we can more clearly see the seasonality for the period from January 1 2009 to February 1 2009. The red lines occur on Mondays (I checked a 2009 calendar), so we can count datapoints to the left or right to see the values for different days of the week. It appears that there are two peaks in each week, one on Saturday (two points to the left of the Monday-lines) and another on Tuesday (one point to the right of the Monday-lines). There are two noticeable troughs in each week, one occurring on Sunday (one point to the left of the Monday-lines) and another on Thursday (three points right of the Monday-lines).



As mentioned, the impact of this seasonality seems very small, compared to the transformed data. The results of the previously performed augmented Dicky-Fuller test indicate that our data is indeed stationary. For what its worth, in developing this report, I did attempt to address the seasonality by removing the value for each datapoint's seasonality from the overall value for that date. This did remove the seasonality component (seasonality was still evident, but the magnitude was reduced to sextillionths), but when I performed a new augmented Dicky-Fuller test, the results were actually slightly worse in terms of the dataset's stationarity. As such, I elected not to try to pursue eliminating this very minor seasonality impact.

Also worth noting is that the apparent seasonality element here, though small, could lend itself to a SARIMAX model, rather than an ARIMA model. I will not be doing this, because the rubric specifies an ARIMA model be used, and I have previously had projects returned to me for "doing too much" and violating the specific requirements of the rubric in similar ways. Thus, we're sticking with an ARIMA model, even if a SARIMAX model might be better.

Trends

>

**Sources Cited:**

Statistics Solutions. (n.d.). Time Series Analysis. Retrieved June 6, 2023, from [https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/time-series-analysis**/**](https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/time-series-analysis/)

**>Source for dicky fuller**

**>Source for seasonality of time series**