# ****Part I:  Research Question****

## A1: RESEARCH QUESTION

This analysis aims to answer the research question: How many days into the future is it possible to reliably forecast the hospital chain’s revenue using time series modeling techniques?

## A2: OBJECTIVES AND GOALS

Goal 1: Develop a reliable [Seasonal] Autoregressive Integrated Moving Average ([S]ARIMA) model using the hospital chain’s daily revenue

* Objective 1: Determine whether the data contains any seasonality
* Objective 2: Determine whether the data contains any trends
* Objective 3: Determine whether any autocorrelation exists within the dataset
* Objective 4: Train the [S]ARIMA model accounting for objectives 1-3

Goal 2: Determine the longest possible accurate forecast period for the model in Goal 1

* Objective 1: Create a 30-day forecast using the model from Goal 1
* Objective 2: Measure the accuracy of the forecasted values for each day
* Objective 3: Identify where the accuracy of the forecast drops below acceptable levels

# ****Part II:  Method Justification****

## B: SUMMARY OF ASSUMPTIONS

When modeling a time series, two primary assumptions are made: stationarity and autocorrelation. A time series is stationary if the mean and variance of the dependent variable do not change with time (Palachy, 2019). Trends upward or downward and differences that grow or shrink over time are not stationary. Autocorrelation means the values in the data are correlated with one or more previous values (Smith, 2022). For example, if a salesperson’s commission is always low when it was high the week before, the data is said to be autocorrelated.

# ****Part III:  Data Preparation****

## C1: LINE GRAPH VISUALIZATION

Chart, line chart, scatter chart

Description automatically generated

## C2: TIME STEP FORMATTING

The realization has a time step of one day. There are 731 values representing 731 days occurring between 01/01/2020 and 12/31/2021. Since there are no missing values and every day is accounted for, it can safely be stated that there are no gaps in the realized time series.

## C3: STATIONARITY

To evaluate the stationarity of the time series, the Augmented Dickey-Fuller (ADF) test was used with an alpha of .05. This test uses the null hypothesis that the data is a random walk or has a trend. If the absolute value of the test statistic is greater than the absolute value of the t-critical value, the data can be said to have weak stationarity, which is sufficient for [S]ARIMA modeling (Prabhakaran, 2019). Below are the results of the ADF test. The absolute value of the test statistic is approximately 2.22, which is not greater than the absolute value of t-critical for an alpha of .05, of approximately 2.87. This means that the null hypothesis that there is a trend in the data cannot be rejected, and the data is not stationary.

Text

Description automatically generated

The first difference of the time series was taken, meaning each value was subtracted from the next, to coerce stationarity. The differenced data was then tested again using ADF. The new results, below, indicate that the differenced data is stationary. The absolute value of the test statistic, approximately 17.37, is much larger than the absolute value of the critical value for an alpha of .05, which is still approximately 2.87.

Text

Description automatically generated

## C4: STEPS TO PREPARE THE DATA

To prepare the medical\_time\_series data for analysis, the following steps were taken:

1. The file was loaded into a pandas DataFrame and previewed.

Text

Description automatically generated

1. The DataFrame was checked for duplicate values.



1. The DataFrame was checked for missing values.



1. The DataFrame was converted to a time series object.

Text

Description automatically generated

1. The summary statistics were reviewed.

Text

Description automatically generated

1. The realized time series was plotted. See section [C1](#_C1:LINE_GRAPH_VISUALIZATION) for the plot.
2. The stationarity was evaluated. See section [C3](#_C3:STATIONARITY) for screenshots.
3. The data was split into training and testing sets. One month of the time series was reserved for testing and the remainder was used for training.
4. The training and testing data sets were saved to Comma Separated Values files for inclusion with this submission.

## C5: PREPARED DATASET

The cleaned and prepared data has been included with this submission as “training\_data.csv” and “testing\_data.csv”.

# ****Part IV:  Model Identification and Analysis****

## D1: REPORT FINDINGS AND VISUALIZATIONS

To inspect the training data for seasonality, it was first de-trended by subtracting the rolling 180-day mean. A period of 180 days was selected because many organizations operate on quarterly financial periods, which are 90 days. The autocorrelation of the detrended data was then plotted. If seasons were present, there would be correlation spikes at regular intervals in the plot. As can be seen, this is not the case, meaning the data does not have a seasonal component.

A picture containing table

Description automatically generated

The auto\_arima function was used to confirm the lack of seasonality and to help determine the best ARIMA model parameters.

Table

Description automatically generated

To assess the trends in the training data 30-, 90-, 180-, and 365-day rolling means were calculated over the daily revenue. In all cases, the trend was positive overall. In the longer trends (180- and 365-day) the data appeared to trend negatively in the last quarter.

Chart, line chart

Description automatically generated

Autocorrelation and partial autocorrelation were plotted after differencing the training data. The plot confirms what was seen in the auto\_arima run, that the best model is AR(1) of the differenced data or ARIMA(1,1,0).

Chart, line chart

Description automatically generated

A power spectral density plot was used to view the periodicity of the data. The detrended data was used to create the plot to prevent trends from obscuring the potential periods. There are spikes and dips in the plot, however, none of them appear at regular intervals which could be considered periodic. This confirms the data is not seasonal.

Chart, line chart, scatter chart

Description automatically generated

For due diligence, the seasonal decomposition was also plotted. A period of 90 days was used because it is a typical revenue cycle for hospitals. In the plot below, the original data, the trend, the seasons, and the residuals can all be seen. The trend appears to capture the majority of the movement in the data while the seasons appear to account for less than 5% of the movement. This is to be expected since the data was already determined not to be seasonal.

Chart

Description automatically generated

The residuals of the seasonally decomposed data were assessed to confirm no trend remained. When applying the rolling 90-day rolling mean, no obvious upward or downward trends were remaining which indicates the decomposition captured them all. There does, however, appear to be an almost 90-day period introduced into the residuals. This is also expected since the data did not naturally have seasons. It would make sense that forcing 90-day seasonal periods would also force the same seasonality on the residuals. This further confirms the lack of seasonality in the data.

Chart, line chart

Description automatically generated

## D2: ARIMA MODEL

Using the results of the model selection phase, an ARIMA(1,1,0) model was selected. The AR(1) portion of the model captures the autoregressive nature of the data while the integrated (I) portion of the model handles taking the first difference of the data to account for the noted upward trend. The Prob (Q) of 0.96 and Prob(JB) of 0.45 indicate that the residuals are both normally distributed and not autocorrelated.

Table

Description automatically generated

Plotting the model’s diagnostics shows that no trends or seasonality appear in the residuals. They truly appear to be random. The kernel density estimate (KDE) shows that the distribution is very close to normal (N(0,1)). The Q-Q plot also shows that the residuals are normally distributed. The correlogram shows that the residuals do not have any remaining autocorrelation. All of these plots indicate that the model should produce reliable forecasting results.

Chart, histogram

Description automatically generated

## D3: FORECASTING USING ARIMA MODEL

Below is the 30-day forecast along with the training and testing data. The first plot displays the training data ranging from January 1st, 2020, to November 30th, 2021, and the testing data ranging from December 2021 through the end of the year, both in blue. The 30-day forecast is displayed in red, and the 95% confidence interval is displayed in pink. The bottom plot omits the training data to zoom in on the forecasted values. There it can be seen that the observed values consistently fall between the lower and upper bounds of the 95% confidence interval.

Chart, line chart, histogram

Description automatically generated

## D4: OUTPUT AND CALCULATIONS

The model was evaluated using the mean absolute error calculated as the mean of the absolute value of the residuals.

When calculating MAE over the training data, the result was 0.35. This means that, on average, the in-sample predicted values were ± $0.35 million from the observed values.



To determine how far out the hospital chain’s revenue could reliably be predicted, the same calculations were performed over the 30-day forecast. For each day in the forecast period, the cumulative MAE was calculated. The cumulative MAE was then calculated as a percentage of the revenue for the last day in each forecast period. The following table captures the results.

Table

Description automatically generated

The error, cumulative MAE, and cumulative MAE as a percent of Revenue can be seen in the following visualizations.

Chart, line chart

Description automatically generated

## D5: CODE

The following code was extracted from “task1.py” which has been included with this submission.

**import** numpy **as** np  
**import** pandas **as** pd  
**from** pmdarima **import** auto\_arima  
**from** matplotlib **import** pyplot **as** plt  
**from** statsmodels.tsa.seasonal **import** seasonal\_decompose  
**from** statsmodels.tsa.stattools **import** adfuller  
**from** statsmodels.tsa.arima.model **import** ARIMA  
**from** statsmodels.graphics.tsaplots **import** plot\_acf, plot\_pacf

*# Import data*

df = pd.read\_csv(**'medical\_time\_series .csv'**, index\_col=**'Day'**)

*# Convert DataFrame to timeseries object (Derek O, 2020).*

df.index = pd.to\_datetime(df.index, unit=**'D'**,  
 origin=pd.Timestamp(**'2019-12-31'**)).to\_period(**'D'**)

*# ADFuller to test for stationarity*adf = adfuller(df, autolag=**'AIC'**)  
adf\_df = pd.Series(adf[:4], index=[**'Test Statistic'**, **'p-value'**,  
 **'Lags'**, **'Observations'**])  
**for** k, v **in** adf[4].items():  
 adf\_df[**f'Critical Value {**k**}'**] = v  
print(**f'Augmented Dickey-Fuller test:\n{**adf\_df.to\_string()**}\n'**)

*# Take the difference to coerce stationarity*print(**'Differencing the data to coerce stationarity...'**)  
df\_diff = df.diff().dropna()  
  
*# Verify stationarity*adf = adfuller(df\_diff, autolag=**'AIC'**)  
adf\_df = pd.Series(adf[:4], index=[**'Test Statistic'**, **'p-value'**,   
 **'Lags'**, **'Observations'**])  
**for** k, v **in** adf[4].items():  
 adf\_df[**f'Critical Value {**k**}'**] = v  
print(**f'Augmented Dickey-Fuller test:\n{**adf\_df.to\_string()**}\n'**)

*# Split the pre-differenced data saving last month for test*df\_train = df.loc[:**'2021-11-30'**]  
df\_test = df.loc[**'2021-12-01'**:]

*# Check for seasonality*detrend = df\_train - df\_train.rolling(180).mean()  
detrend.dropna(inplace=**True**)  
fig, ax = plt.subplots(figsize=(10, 3))  
fig.suptitle(**'Detrended Daily Revenue'**)  
plot\_acf(detrend, ax=ax, lags=180, zero=**False**)  
plt.tight\_layout()  
plt.show()

*# Auto-ARIMA to get p, d, q and check for seasonality*stepwise\_fit = auto\_arima(df\_train.Revenue, trace=**True**,   
 suppress\_warnings=**True**)

*# Inspect trends*data = {}  
trends = [30, 90, 180, 365]  
\_, ax = plt.subplots(figsize=(10, 3))  
**for** t **in** trends:  
 data[t] = df\_train.rolling(t).mean()  
 data[t].columns = [**f'Rolling {**t**} Day Mean'**]  
 data[t].plot(ax=ax, grid=**True**)  
ax.set\_title(**'Trends'**)  
plt.tight\_layout()  
plt.show()

*# Perform acf/pacf*df\_train\_diff = df\_train.diff().dropna()  
\_, (ax1, ax2) = plt.subplots(2, figsize=(10, 6), sharey=**True**)  
plot\_acf(df\_train\_diff, zero=**False**, ax=ax1)  
plot\_pacf(df\_train\_diff, zero=**False**, method=**'ywm'**, ax=ax2)  
plt.tight\_layout()  
plt.show()

*# Perform spectral density to see the periodicity*\_, ax = plt.subplots(figsize=(10, 3))  
ax.psd(detrend.Revenue)  
ax.set\_title(**'Spectral Density of Detrended Revenue'**)  
plt.tight\_layout()  
plt.show()

*# Decompose to see components of time series*decomp\_results = seasonal\_decompose(df\_train.Revenue, model=**'additive'**,   
 period=90)

fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, figsize=(10, 12), sharex=**True**)  
fig.suptitle(**f'Seasonal Decomposition (period=90)'**)  
decomp\_results.observed.plot(grid=**True**, ax=ax1, title=**'Original'**)  
decomp\_results.trend.plot(grid=**True**, ax=ax2, title=**'Trend'**)  
decomp\_results.seasonal.plot(grid=**True**, ax=ax3, title=**'Seasons'**)  
decomp\_results.resid.plot(grid=**True**, ax=ax4, title=**'Residuals'**)  
plt.tight\_layout()  
plt.show()

*# Confirmation of lack of trends in residuals*mean\_residuals = decomp\_results.resid.rolling(90).mean()  
mean\_residuals.dropna(inplace=**False**)  
\_, ax = plt.subplots(figsize=(10, 3))  
decomp\_results.resid.plot(ax=ax, label=**'Residuals'**)  
mean\_residuals.plot(ax=ax, label=**'Rolling 90-day Mean Residuals'**,   
 grid=**True**, title=**'Residuals'**)

plt.legend()  
plt.tight\_layout()  
plt.show()

# ****Part V:  Data Summary and Implications****

*# Create and fit ARIMA(p,d,q) model*mod = ARIMA(df\_train, order=(1, 1, 0))  
res = mod.fit()  
print(res.summary(), **'\n'**)

*# Evaluate the model*mae = np.mean(np.abs(res.resid))  
print(**'Mean Absolute Error:'**, mae, **'\n'**)  
res.plot\_diagnostics(figsize=(9, 6))  
plt.tight\_layout()  
plt.show()

*# Predict with the test data*forecast = res.get\_forecast(30)  
ci = forecast.conf\_int()  
\_, (ax1, ax2) = plt.subplots(2, figsize=(10, 6))  
df.plot(ax=ax1, color=**'b'**)  
forecast.predicted\_mean.plot(ax=ax1, color=**'r'**, label=**'Forecasted Revenue'**)  
ax1.fill\_between(ci.index, ci[**'lower Revenue'**], ci[**'upper Revenue'**],  
 color=**'r'**, alpha=0.2, label=**'95% Confidence Interval'**)  
ax1.set\_title(**'Observed Revenue with 30-day Forecast'**)  
df\_test[:**'2021-12-30'**].plot(ax=ax2, color=**'b'**)  
forecast.predicted\_mean.plot(ax=ax2, color=**'r'**, label=**'Forecasted Revenue'**)  
ax2.fill\_between(ci.index, ci[**'lower Revenue'**], ci[**'upper Revenue'**],  
 color=**'r'**, alpha=0.2, label=**'95% Confidence Interval'**)  
ax2.set\_title(**'30-day Observed vs Forecasted Revenue'**)  
plt.legend()  
plt.tight\_layout()  
plt.show()

*# Evaluate forecast mean absolute error for 1-30 days*metric\_df = df\_test.join(forecast.predicted\_mean)  
metric\_df[**'resid'**] = metric\_df.predicted\_mean.sub(metric\_df.Revenue)  
metric\_df[**'cum\_mae'**] = metric\_df.resid.abs().cumsum().div(metric\_df.index.day)  
metric\_df[**'cum\_mae\_pct\_Revenue'**] = metric\_df.cum\_mae.div(metric\_df.Revenue)  
print(metric\_df.to\_string())  
  
fig, (ax1, ax2, ax3) = plt.subplots(3, sharex=**True**, figsize=(10, 10))  
fig.suptitle(**'30-day Forecast Metrics'**)  
metric\_df.resid.plot(ax=ax1, grid=**True**, marker=**'o'**, title=**'Error'**)  
metric\_df.cum\_mae.plot(ax=ax2, grid=**True**, marker=**'o'**,   
 title=**'Cumulative Mean Absolute Error'**)  
metric\_df.cum\_mae\_pct\_Revenue.plot(ax=ax3, grid=**True**, marker=**'o'**,  
 title=**'Percent Cumulative Mean '  
 'Absolute Error of Revenue'**)

plt.tight\_layout()  
plt.show()

## E1: RESULTS

Once the data was prepared, the analysis began with model selection. To select the best [S]ARIMA model, several components of the time series were evaluated. First, the integrated (I) parameter was determined by evaluating the dataset stationarity using the Augmented Dickey-Fuller (ADF) test. The resultant p-value was greater than .05, which did not allow the null hypothesis that the data is not stationary to be rejected, meaning the data was not stationary. The first difference was taken to coerce stationarity. After running the ADF test on the first difference time series, stationarity was observed with a resultant p-value less than .05, indicating that the results of the test were significant. With the trends identified and accounted for, the presence or lack of seasonality was assessed to determine the need for a seasonal,[S], model. To check for seasonality, the pre-difference data were detrended by subtracting the 180-day rolling mean from the time series. A period of 180 days was used because a typical revenue cycle in hospital systems is 90 days or one-quarter of a year. The de-trended data was assessed for autocorrelation using a correlogram with 180 lags. If seasons were present in the data, there would have been one or more repeating spikes in correlation, presumably at regular intervals. The plot displayed no such pattern, so the data was determined to not have a seasonal component. This was confirmed using the auto\_arima function, which iteratively evaluates the models produced for all hyperparameters of the SARIMA model. The final model parameters were ARIMA(1,1,0)(0,0,0)[0]. The first set of parentheses indicates that the autoregressive (AR) portion should be 1, the integrated (I) portion should be 1, and the moving average (MA) portion should be 0. The second set of parentheses indicates that there is no seasonal AR, I, or MA portions of the model. The final bracket indicates that there is no seasonal period. Overall, these results indicate an AR(1) model should be used after taking the first difference of the time series. To confirm the AR and MA portions of the model, correlograms were created for both the autocorrelation and the partial autocorrelation functions. The patterns displayed confirmed that the AR(1) model was most appropriate.

The original data was collected using a 1-day interval. This level of granularity can capture daily, weekly, monthly, quarterly, and yearly trends and seasons in the data. It does not, however, lend itself well to long-term forecasts because the data is more volatile at higher granularities, containing far more noise than data collected on the last day of each month, for instance. Since it was found that there were no defined seasons present in the data, there would not have been added value in trying to forecast out further to capture them. Additionally, since the research question aimed to see how many days into the future the revenue could reliably be forecasted, the final interval of 1 day was used for forecasting. As was defined in the research goals and objectives, a 30-day forecast length was used. It was not expected that the forecast would be reliable in or further than one month since anything forecasted beyond one day would have to be produced dynamically. Predictions such as this tend to become less reliable over time since the data on which predictions are made are themselves predicted values.

The metric used during model selection was the Akaike information criterion (AIC), which measures how well a model fits its training data. It is not useful in isolation since it is not bounded. However, when comparing two or more models, it can be used to find the model with the best fit. A lower AIC value indicates the model fits the data better than a model with a higher AIC value. Once the model was selected, the Mean Absolute Error (MAE) was used to measure the cumulative MAE for each forecast length between 1 and 30 days. This metric is also not bounded but does directly translate to meaningful information. For instance, when evaluating the overall MAE on the training data, the value was 0.35. This means that, on average, the in-sample one-day forecast is within ±$.35 million of the observed revenue. When extrapolating the model beyond the training data, forecasting values 1 to 30 days in the future, the cumulative MAE ranged from 0.88 to 2.05. It can be difficult to conceptualize error measured in millions of U.S. Dollars, so it made sense to evaluate the metric as a proportion of the total revenue at the end of the predicted period. This allows executives to understand the accuracy of forecasted data as it relates to the revenue obtained during that period. Using the forecasted values and the calculated metrics, a few findings were present. First, the predicted mean of the forecast flatlines beginning on December 15, 2021, or at two weeks. As was stated earlier, it is to be expected that the values become less accurate the farther they are predicted into the future. This indicates that no predictions beyond 2 weeks, or 14 days, should be relied upon. The second finding was that the highest cumulative MAE occurred at 10 days with a value of ±$2.05 million or 10.38% of the revenue at the end of the forecast period. Depending on the operating margins of the hospital chain, a more conservative forecast period should be used. In the 1- and 2-day forecasts, the margin of error is below 5%. To be safe, no longer than 7 days, with a cumulative MAE percent of 8.4%, should be predicted using the model with the currently collected data.

## E2: ANNOTATED VISUALIZATION

The following visual, also displayed in section [D3](#_D3:FORECASTING_USING_ARIMA), depicts the 30-day forecast and 95% confidence intervals, using the identified ARIMA model compared to the final month of data that was reserved for testing.

Chart, line chart, histogram

Description automatically generated

## E3: RECOMMENDATIONS

The model should be used in an exploratory fashion to continue model evaluation for 7-day forecasts. Though, without model updates, it is likely that the model will not maintain any level of accuracy for long in the beginning. This leads to the second recommendation, which is to collect more data with which the model can be updated. This is needed for two reasons. First, time series models are more accurate when supplied with recent data. Second, the first year of data is reflective of the organization’s state as a startup. The trends appear to normalize in the second year. With continued evaluation, seasons may become visible, allowing the model to forecast more accurately, and farther into the future. The last recommendation is to automate the process of staging the most recent data and updating the model. By automating this step, the model will always have access to the most recent data, allowing data science efforts to focus on tuning the model for the best performance rather than obtaining more training data.

# ****Part VI:  Reporting****

## F: REPORTING

The HTML export of the Jupyter Notebook report can be found included with this submission as “task1.html”.

## G: SOURCES FOR THIRD-PARTY CODE

Derek O. (2020, May 30). *python – Ordering Timestamps for ARIMA model prediction.* Retrieved from Stack Overflow: https://stackoverflow.com/a/62098606/15139623

H: [SOURCES](https://lrps.wgu.edu/provision/147882373)

Palachy, S. (2019, April 8). *Stationarity in time series analysis*. Retrieved from Towards Data Science: https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322

Prabhakaran, S. (2019, November 2). *Augmented Dickey Fuller Test (ADF Test) - Must Read Guide*. Retrieved from ML+: https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

Smith, T. (2022, May 28). *What Is Autocorrelation*. Retrieved from Investopedia: https://www.investopedia.com/terms/a/autocorrelation.asp