Eli Bolotin
DS700: Final Project
July 20th, 2018

Fargo Health Group Case:
Modeling and Forecasting Demand

Case summary

The Fargo Health Group (FHG) case discusses a problem that is common in business to every industry. At a high-level, the case is about an organization that is struggling to meet customer demand. In particular, FHG is struggling to meet the demand of disability examinations due to lack of examining physicians. As a result, Fargo's health centers "are late [in] completing the requested examinations" and the organization is incurring expenses and fees. In the words of Jay Rubin, Fargo's director of the Quality Assessment Office (QAO), "the organization needs to reduce the costs and fees paid" to outpatient clinics and the Regional Office of Health Oversight [2]. To reduce such expenses, Fargo needs to implement data-driven planning to forecast demand and allocate physicians accordingly. The goal is to develop predictive analytics to better manage demand, improve scheduling, and reduce expenses [3].

Approach & Strategy

The purpose of this project is to achieve the stated goal by modeling historical demand. Data is collected on "34 Fargo clinics and for each examination type" [2], and we will use a sample of this data to train a model that is representative of a more general population – meaning all health centers. The sample dataset contains incoming examinations for cardiovascular exams beginning in 2006 through the end of 2013 at the Abbeville, LA health center (HC). Our model must "learn from [Fargo's] historical request volumes" to predict future incoming examinations and improve scheduling. We have identified several characteristics of the sample data that determine the model that is created, as well as the preparation that is needed to utilize this data in our model:

1) First and foremost, the sample data is a *time series* from 2006-01 to 2013-12.
2) Secondly, the sample has "quality issues" such as incomplete data, missing data, incorrect data types, and outliers.

As a result of the first point above, our model must consider time series properties such as trend, seasonality, and irregularity. Therefore, we have several options to consider in deciding how to model the data. We can use simple moving averages, exponential forecasting, and ARIMA forecasting to models to "explain" our data and make predictions. More will be said on this later. First, we must clean the dataset so that it can be utilized for forecasting. The overall process to solving this case can be separated into three stages: data cleaning, data imputation, and forecasting. They are explained in detail.

Stage 1: Data Cleaning

Part I: Categorization – The sample dataset of the Abbeville, LA contains 96 observations consisting of 3 fields: incoming examinations (response variable), year and month. Of 96 observations, 15 had issues of the following types: data type error, missing values, outliers, or incomplete. So, the first step is to categorize observations that need cleaning. In the attached Excel workbook titled *Dataset.xlsx*, the worksheet 'Abbeville, LA_v3' shows the highlighted categorized observations in columns B:C (legend in column G).

Next, we need to decide what action to take with regard to the observations that need cleaning. As indicated in the *Explanation of Dataset.pdf* file, some of the observations with erroneous values can be fixed. Those that cannot be fixed will be imputed. So, in column D, we classify our erroneous observations into one of two "actions". The possible actions include 'Calculate' or 'NA'. 'Calculate' means to fix the incorrect value by computing it from given assignment information. 'NA' means to leave the incorrigible value for the next stage of imputation.

Part II: Fixing erroneous data – As mentioned above, erroneous values were identified and categorized on the basis if they could be fixed or had to be imputed. These values were marked 'Calculate' (fix) or 'NA' (impute). The incorrigible values marked 'NA' were either data type errors or outliers. A total of 8 observations were marked for imputation.

Observations & values that could be amended were discovered through details in *Explanation of Dataset.pdf* file. Correctable values were caused by 5 underlying problems. Below is a brief explanation of how each problem was addressed (reference the worksheet 'Abbeville, LA_v3'):

1) Problem 1 – the correct number of exams for May 2007 was recovered by counting how many exams were rerouted to other HCs (G116:H122).
2) Problem 2 – the value of incoming examinations for October 2008 was marked for imputation because it was an irregular outlier with negative value to our model. This was due to the fact that this outlier was an anomaly caused by a hurricane and was not useful to increasing the accuracy of our model.
3) Problem 3 – to allocate the 5129 missing exams between the 3 months of Dec-2009 through Feb-2010, a weighted approach was used. Averages of examinations were produced for Dec, Jan, and Feb for every available year in the dataset (L87:L90).
4) Problem 4 - the correct number of exams for May, June, and July of 2013 was recovered by counting how many were rerouted to other HCs (G32:K42).
5) Problem 5 - the correct number of exams for December 2013 was recovered by analyzing data in the 'December 2013 data' worksheet and filtering the data to find rerouted exams by heart-related conditions (H17).

Stage 2: Data Imputation

Part I: Considerations and process – As described above, at this point our dataset has been updated with 7 cleaned values and has a remaining of 8 observations marked for imputation. An alternative to imputing these values would be to delete the observations altogether (called listwise deletion). In our case, it is not optimal to pursue listwise deletion because this would amount to an 8.33% reduction in the sample size, resulting in decreased model accuracy caused by parameter bias [4]. Therefore, the favored approach is multiple imputation (MI), whereby "a set of complete datasets is generated from an existing dataset" [5] and the missing values are filled, analyzed, and combined using statistical methods. The two R imputation packages that we will utilize are called 'Mice' and 'Amelia'. The general steps of imputation that I have followed are as follows:

1. Import the half-cleaned dataset (titled *dataset_for_impute.csv*).
2. Analyze the dataset for missing data.
3. Impute, fit, pool (combine fit) and analyze the imputed data.
4. Compare the imputed data to original values.
5. Plot the imputed data.

Part II: Imputation with Mice – Mice runs multiple imputations (default is 5) on the original data frame and generates a complete dataset for each imputation [5]. I chose to run 10 imputations with 10 iterations on the dataset in hopes of generating a better model. Judging from figures 2 and 3 below, the imputed datasets fit the original data quite well (in the sense their distribution is very similar), which raises confidence for forecasting. Additionally, speaking to figure 4, although our model does not appear linear, it in fact could be modeled with a linear model that has enough parametric terms. After fitting a linear model on each imputation (commented in R), and pooling and summarizing the results, our p-values are significant which indicates that the null hypothesis can be rejected and that our parameters "explain" correlation. As such, due to the accuracy of the model and ideal distribution of imputations, we are able to confidently use *Mice* imputation for the purpose of forecasting.

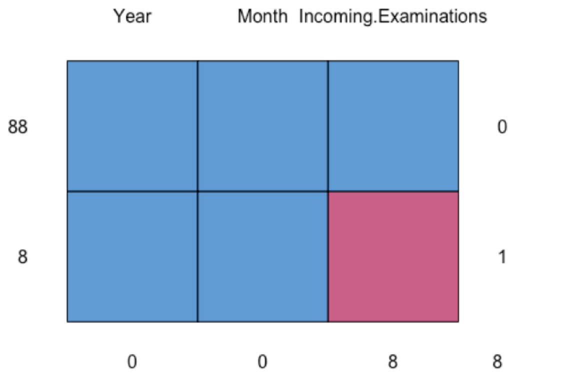


Figure 1 – Plot of missing data patterns showing many dataset rows (left axis) are missing how many values (right-axis) and which variables are missing them (bottom axis)

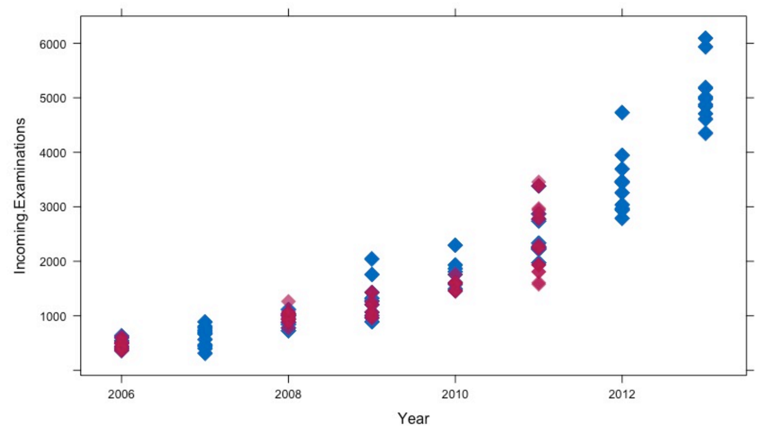


Figure 2 – Xyplot showing how the imputed values for examinations (in red) fall in comparison to the original data.

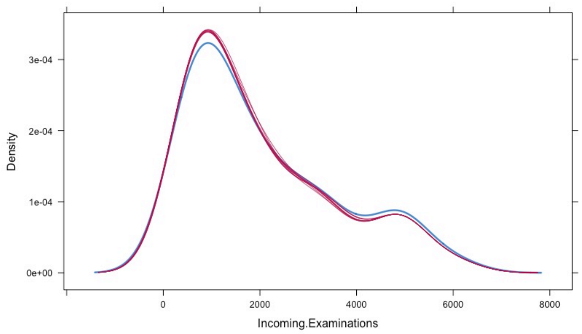


Figure 3 – Density plot showing distribution of imputed datasets (red) compared to original dataset (blue)

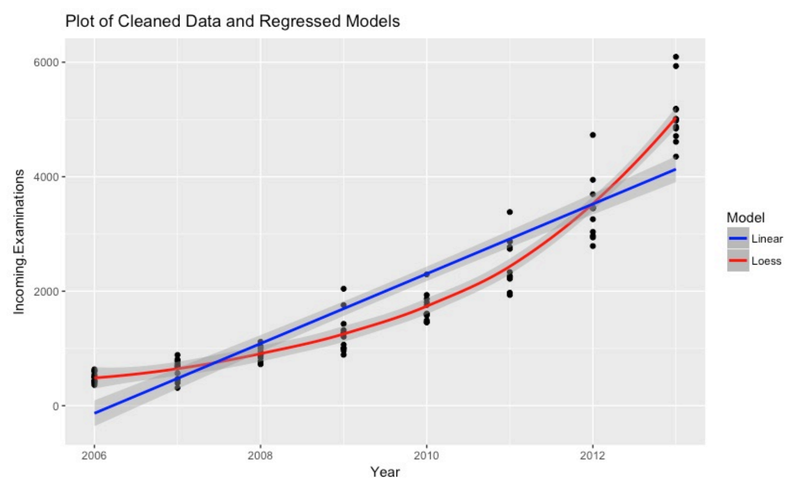


Figure 4 – Scatter plot of examinations fitted with 2 models: a linear model and Loess model (nonparametric).

Part III: How does Amelia compare to Mice? – The process of imputation with *Amelia* is similar to *Mice*, but with several notable technical differences. After imputing the data with *Mice* and fitting a regression to each imputed set, it is then possible to "pool" the regressed models into one combined form. According to R documentation, pooling the models is equivalent to "averaging the estimates of the complete data model and [computing] the total variance over the repeated analyses" [6]. The pooled summary provides a combined regression for all of the imputations. To arrive at a similar combined regression in *Amelia*, it is first necessary to transform the data from a list into a data frame.

Imputation with *Amelia* produces data that is similarly distributed to actual observations (figure 5). In figure 6, the residuals vs leverage plot shows that the regression is minimally influenced by "outlying" data points. In figure 7, the Q-Q plot shows that *Amelia's* fit residuals follow a *mostly* normal distribution, but not absolutely. Figure 8 confirms that the residuals of fitted (estimated) values do not follow a linear pattern. Numerous parametric terms will be required to fit the data adequately. Figures 9 and 10 show that the summary statistics on *Amelia* and *Mice* models are quite similar. In both models, the high significance of the coefficients, p-values, and F-statistics show that the models explain the response variable beyond chance levels and that a relationship exists between the response variable and predictor variables, suggesting that the null hypothesis should be rejected. Lastly, the high multiple & adjusted R-squared in both the models indicate that they explain ~76% and ~85% (respectively) of total error while adjusted for number of terms. Figure 11 shows the adjusted R-squared for the combined *Mice* imputed datasets. This statistic represents how well the model explains variance while accounting for the number of terms. The R-squared of ~86% in figure 11 is very close to the same statistic for a single imputed dataset as shown in Figure 10. Finally, in Figure 12 we see the pooled regression for *Mice*, which considers each regression on 10 imputed datasets. Note how similar this regression is to that of a single *Mice* imputed dataset in Figure 10. Every model both on individual and combined imputed datasets for *Mice* and *Amelia* is statistically significant.

Part IV: Conclusion – The modeling results grant confidence to the use of either *Mice* or *Amelia* imputed data in forecasting. So, how do we choose which imputation method to use for supplying a final cleaned dataset for the next stage of forecasting? In my opinion, the first answer is in Figure 11. The adjusted R-squared for *Mice's* pooled regressions is higher than the same statistic for *Amelia*, as seen in Figure 9. Additionally, *Amelia's* imputations have a significantly higher range of residuals. As a result, due to the fact that our sample size is relatively small, the conclusion to use *Mice* for imputation is a safer bet.

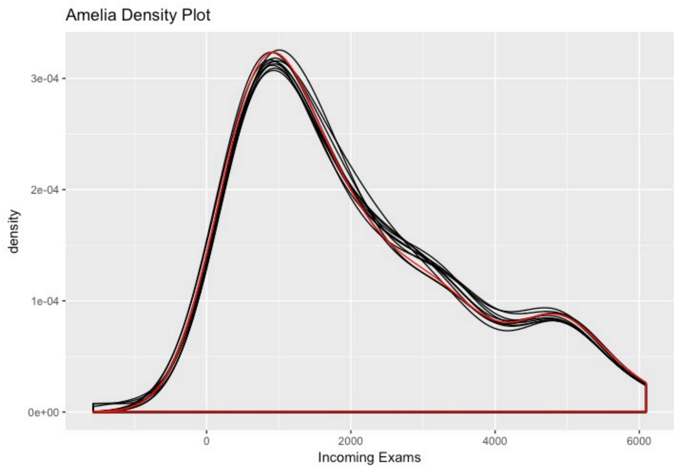


Figure 5 – Density plot of *Amelia's* 10 imputed datasets. The red density plot is that of actual data.

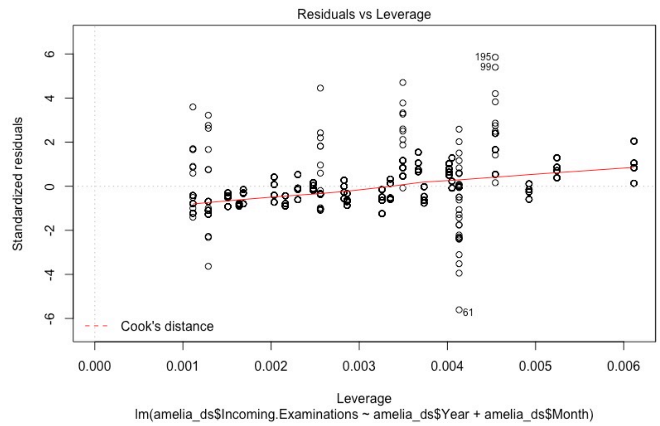


Figure 6 – Residuals vs Leverage plot (*Amelia* fit)

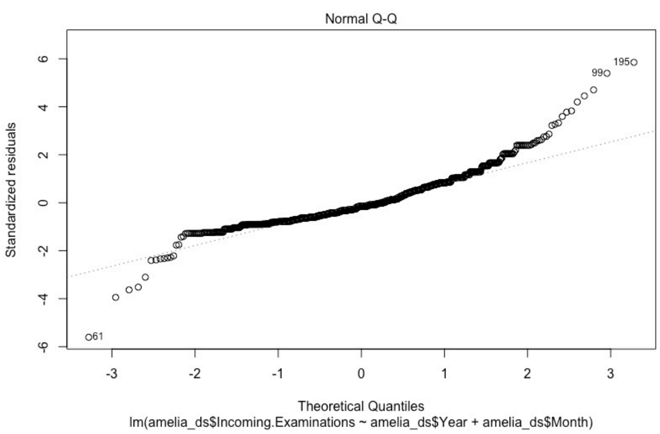


Figure 7 – The Q-Q plot of *Amelia* fit

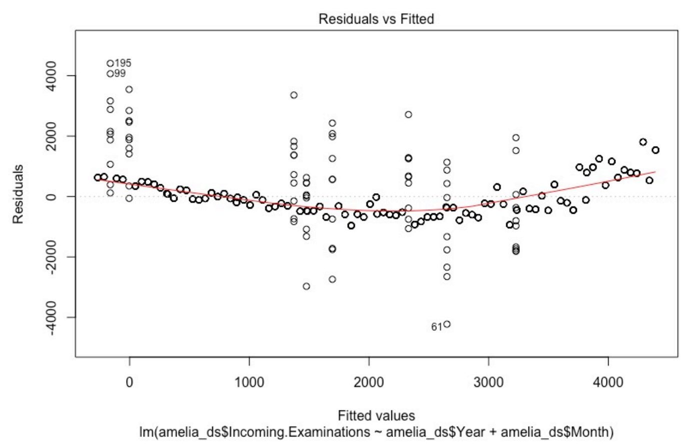


Figure 8 – Residuals vs Fitted values (*Amelia*).

```
Call:
lm(formula = amelia_ds$Incoming.Examinations ~ amelia_ds$Year +
    amelia_ds$Month)

Residuals:
    Min      1Q  Median      3Q     Max
-4222.4  -480.4  -113.4   396.7  4406.5

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.171e+06  2.136e+04 -54.811  < 2e-16 ***
amelia_ds$Year   5.835e+02  1.063e+01  54.892  < 2e-16 ***
amelia_ds$Month  5.239e+01  7.056e+00   7.425  2.5e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 754.7 on 957 degrees of freedom
Multiple R-squared:  0.7622,	Adjusted R-squared:  0.7618
F-statistic:  1534 on 2 and 957 DF,  p-value: < 2.2e-16
```

Figure 9 – Summary function on the linear model on *Amelia's* combined imputed datasets.

```
Call:
lm(formula = Incoming.Examinations ~ Year + Month)

Residuals:
   Min     1Q Median     3Q    Max
-915.6 -458.7 -137.5  380.2 1768.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.228e+06  5.155e+04 -23.818  < 2e-16 ***
Year         6.119e+02  2.566e+01  23.851  < 2e-16 ***
Month        5.252e+01  1.703e+01   3.084  0.00269 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 576 on 93 degrees of freedom
Multiple R-squared:  0.8615,	Adjusted R-squared:  0.8585
F-statistic: 289.2 on 2 and 93 DF,  p-value: < 2.2e-16
```

Figure 10 – Summary output is called on the linear regression of *Mice's* imputed dataset #1.

```
              est     lo 95     hi 95 fmi
adj R^2 0.8579779 0.7942441 0.9031312 NaN
```

Figure 11 – This figure produces the adjusted R-squared for the entire *Mice* imputed dataset collection.

```
                 estimate   std.error  statistic       df    p.value
(Intercept) -1.224184e+06 51633.75773 -23.708984 90.75259 0.00000000
Year         6.100202e+02    25.69475  23.741046 90.75261 0.00000000
Month        5.481914e+01    17.10999   3.203925 90.09275 0.00187097
```

Figure 12 – Summary output of *Mice's* "pooled" fit.

Stage 3: Forecasting

Two types of models were considered for the purpose of forecasting. One of them is ARIMA and other is Holt exponential smoothing. Both approaches and their results are discussed below in detail.

Part I: Overview of ARIMA – In this approach to forecasting, "predicted values are a linear function of recent actual values and recent errors of prediction (residuals)" [8]. This method of forecasting consists of two *integrated* components: AR (autoregressive) & MA (moving average) terms [9]. The AR component is calculated from a "linear combination of previous *p* values" and the MA component is calculated from a "linear combination of *q* previous errors" [11]. ARIMA is a method of forecasting that is designed to fit stationary time series, in which "the statistical properties of the series don't change over time" [10]. A stationary time series *does not* have a trend, *does have* constant variance about the mean, and *does have* constant autocorrelation over time [12]. These requirements of ARIMA are the basis for the process that I took in developing an ARIMA forecast (based on a *Mice* imputed dataset). Below are the steps:

1. Plot the time series and evaluate stationarity.
   a. Make the plot stationary if it is not.
2. Identify number of AR and MA terms (values of p and q).
3. Fit the model on time series.
4. Evaluate model's accuracy, fit, and residuals.
5. Make forecasts and show predictions.

Part II: Results of ARIMA – In summarizing the results, let's begin by plotting the time series of the *Mice* imputed dataset (figure 13). This plot is then decomposed to understand the prevalence of three components: trend, seasonality, and error. The decomposition shows presence of all three components, which indicates that the time series is not stationary. The ACF plot in figure 14 shows a trend of coefficients by lag, and the ADF test confirms stationarity with a highly nonsignificant value at .9884. This means that the null hypothesis that the time series is nonstationary must be accepted. It can be made stationary by being "differenced" against itself. This process makes the chart stationary by removing additive components such as trend and seasonality (by subtracting the difference in Y values at different periods – i.e. Y_t-Y_{t-1}) [13]. We choose to difference twice in order to remove a quadratic trend, which is suggested by seasonality. After differencing, figure 16 indicates that our chart is now stationary, with a p-value of 0.01 (thus, the null hypothesis is rejected).

Next, we must identify how many AR and MA terms to include in our ARIMA model. To do so, the ACF and PACF are necessary. The ACF chart plots autocorrelation, which "measures the way observations in a time series relate to each other" [10]. The PACF chart shows autocorrelation (AC) between two points, but with the AC of all other points in between removed. The lag at which ACF is lowest helps determine the number of error terms (MA or 'q' term), while the lag at which PACF is lowest helps determine the number of predictor terms (AR or 'p' term). We will select these terms for a custom ARIMA model, and then compare the results against an auto-generated ARIMA model. The ACF chart shows that AC cuts off around lag 6, so this can be our 'q' term. The 'p' term appears to be 7, because this is the lag at which the AC cuts off on the PACF chart. Proceeding, we run a summary on both our custom ARIMA fit and the autofit.

Figures 19 and 20 show these results, and when compared to each other, the custom fit appears to be better. This is because when compared to the autofit, the custom fit has a lower S^2 (meaning lower variance), as well as better error measures in nearly every metric (RMSE, MAE, MAPE, and MASE). As we move on to evaluate the models' accuracy through the use of the Q-Q plot (figures 21 and 22), we show that for both models, the data appears *mostly* normally distributed, but this not convincing. For the ARIMA autofit model, the Ljung-Box test shows a very low chi-squared figure along with a very high p-value. These values both indicate the same thing: the null hypothesis of non-relation is to be accepted, suggesting that our AC are uniformly close to zero and that our predictor variables should adequately explain variance. Interestingly, the Box test resulted in less non-significance for the custom ARIMA fit. This is despite the fact that the custom ARIMA model displays better accuracy in terms of reported error (figure 20). However, I believe this is because the custom ARIMA model has higher mean error. Moving on, figure 23 shows a summary of residuals for both models. The autofit model has a greater range of residuals. Figure 24 plots both models against the time series, showing excellent fit in both cases.

Figures 25 and 26 plot the actual forecasts from the autofit and custom ARIMA models, respectively. A few observations can be made about the differences: the custom fit appears to fit the data better than the autofit, it has a steeper climb than the autofit (check Y-axis), as well as a lower range of distribution in the 95% quantile. This is due to the added AR and MA terms in the custom fit model. It uses more terms in regressing lagged values of itself. Consequently, the custom model appears to be overfitting the data. The problem is that ARIMA is not a linear regression due to the error terms in its formula, so we can't find adjusted and multiple R-squared to confirm this notion.

So, which is better, the autofit or custom fit ARIMA model? Although the custom fit model has better (lower) error metrics, this may result in overfitting if we apply the model to a much larger population. On the other hand, it may just be the more accurate model. The correlation between fitted custom ARIMA values and the observed time series is at .969. The same figure for auto ARIMA model is .960. They must both be considered as finalists in a predictive model decision.

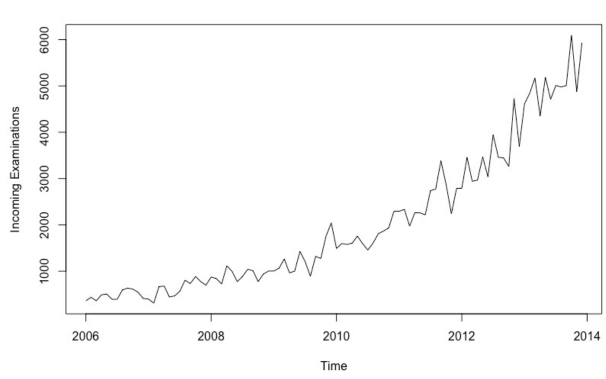


Figure 13 – Time series plot of *Mice* imputed dataset.

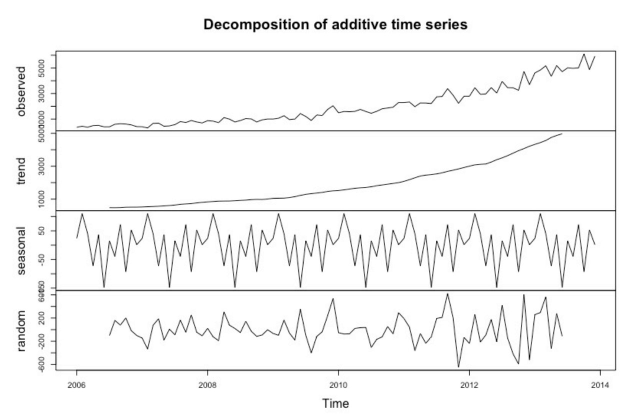
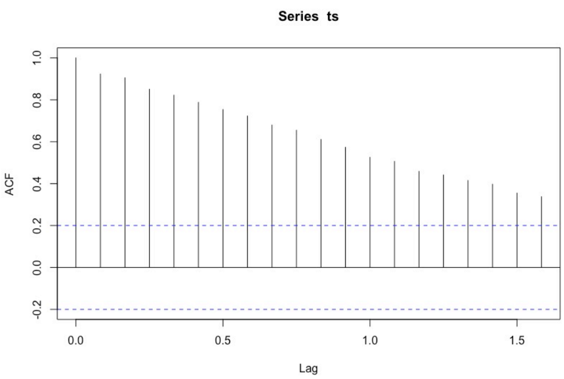
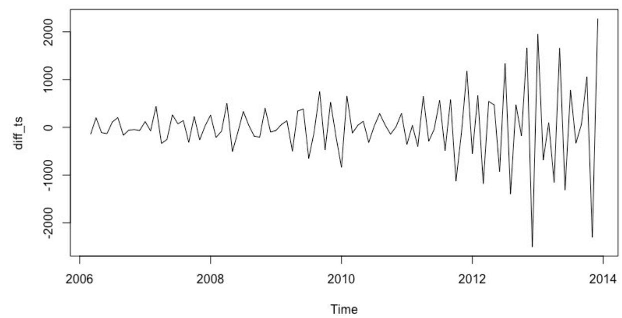


Figure 14 – Decomposition of time series plot in Figure 13.




```
        Augmented Dickey-Fuller Test

data:  ts
Dickey-Fuller = -0.31208, Lag order = 4, p-value = 0.9884
alternative hypothesis: stationary
```

Figure 15 – ACF plot of non-stationary time series. ADF test result below.




```
        Augmented Dickey-Fuller Test

data:  diff_ts
Dickey-Fuller = -8.4792, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Figure 16 – Plot of time series after differencing. ADF test result below.

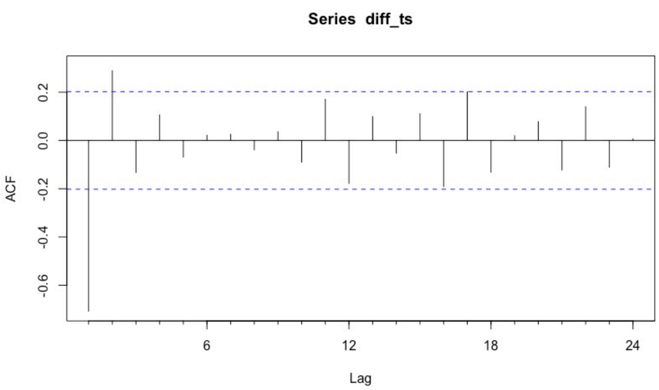


Figure 17 – ACF plot of differenced time series.

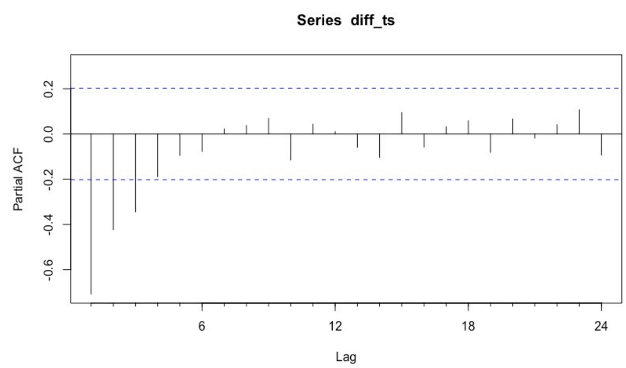


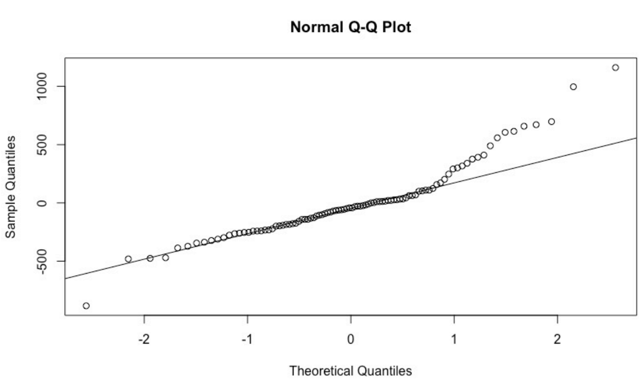
Figure 18 – PACF plot of differenced time series.

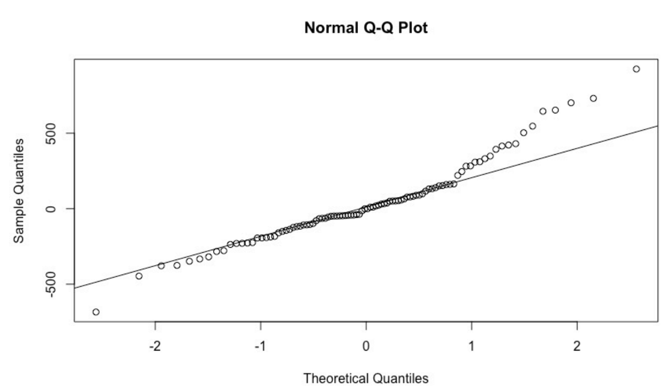
```
Series: ts
ARIMA(1,1,1) with drift

Coefficients:
          ar1      ma1    drift
      -0.2949  -0.4919  54.9360
s.e.   0.1428   0.1282  12.8896

sigma^2 estimated as 103157:  log likelihood=-681.93
AIC=1371.87   AICc=1372.31   BIC=1382.09

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE         ACF1
Training set -0.6192313 314.4183 222.5433 -7.53879 15.81654 0.3324082 9.147416e-05
```

Figure 19 – Summary output of ARIMA auto-generated fit.

```
Series: ts
ARIMA(7,2,6)

Coefficients:
          ar1     ar2    ar3    ar4      ar5      ar6     ar7      ma1      ma2
      -0.5466  0.2136  0.606  0.900  -0.1045  -0.2040  0.1025  -1.3558  -0.1624
s.e.   0.1277  0.1717  0.176  0.111   0.1452   0.1465  0.1300   0.1674   0.3626

sigma^2 estimated as 86696:  log likelihood=-668.53
AIC=1365.06   AICc=1370.37   BIC=1400.66

Training set error measures:
                   ME     RMSE      MAE        MPE     MAPE     MASE        ACF1
Training set 34.12339 270.4629 195.3399 -0.8952236 12.34182 0.291775 -0.03209728
```

Figure 20 – Summary output of ARIMA(7,2,6) fit (some terms cut off in picture).





```
Box-Ljung test

data:  fitauto$residuals
X-squared = 8.2865e-07, df = 1, p-value = 0.9993
```

Figure 21 – Q-Q plot for ARIMA *autofit*.

```
Box-Ljung test

data:  fit$residuals
X-squared = 0.10203, df = 1, p-value = 0.7494
```

Figure 22 – Q-Q plot for ARIMA *custom* fit.

```
> summary(residuals(fit))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-684.2587 -119.7795   -0.0792   34.1234  142.3442  925.1034
>  summary(residuals(fitauto))
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-883.1740 -193.6513  -42.5549   -0.6192  100.6924 1161.3995
```

Figure 23 – Summary output of residuals on both ARIMA models.

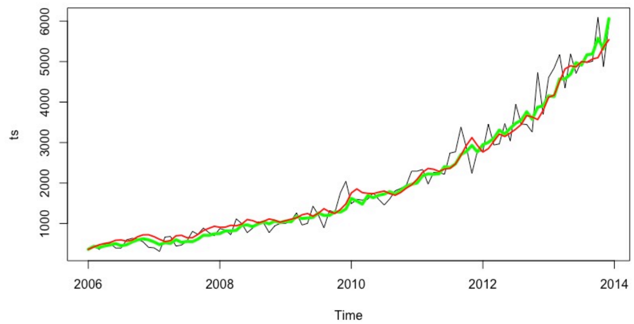


Figure 24 – plot of ARIMA fit (green) and ARIMA auto fit (red) against time series.

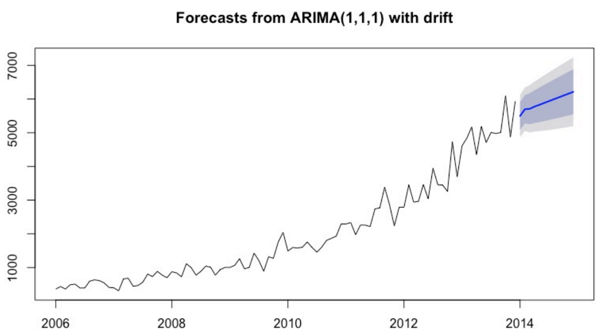


Figure 25 – Plot of ARIMA autofit forecast.

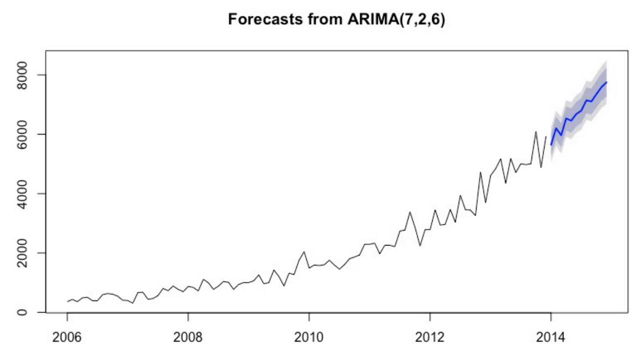


Figure 26 – Plot of ARIMA fit forecast.

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2014	5499.183	5087.574	5910.793	4869.681	6128.686
Feb 2014	5698.272	5277.412	6119.132	5054.622	6341.922
Mar 2014	5710.692	5251.658	6169.725	5008.660	6412.723
Apr 2014	5778.167	5293.639	6262.696	5037.145	6519.190
May 2014	5829.405	5318.063	6340.747	5047.375	6611.435
Jun 2014	5885.432	5349.355	6421.509	5065.573	6705.290
Jul 2014	5940.046	5380.119	6499.974	5083.711	6796.382
Aug 2014	5995.077	5412.333	6577.821	5103.847	6886.308
Sep 2014	6049.985	5445.268	6654.703	5125.150	6974.821
Oct 2014	6104.930	5479.015	6730.844	5147.675	7062.184
Nov 2014	6159.863	5513.444	6806.282	5171.251	7148.475
Dec 2014	6214.800	5548.508	6881.092	5195.794	7233.805

Figure 27 – 12-month forecast according to ARIMA autofit.

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2014	5648.928	5264.640	6033.216	5061.211	6236.646
Feb 2014	6204.625	5818.351	6590.898	5613.870	6795.379
Mar 2014	5966.686	5564.426	6368.946	5351.482	6581.890
Apr 2014	6534.957	6132.134	6937.779	5918.892	7151.021
May 2014	6459.474	6051.289	6867.659	5835.209	7083.738
Jun 2014	6681.212	6267.628	7094.796	6048.689	7313.735
Jul 2014	6797.003	6380.862	7213.145	6160.570	7433.437
Aug 2014	7143.707	6713.717	7573.698	6486.093	7801.322
Sep 2014	7105.961	6670.395	7541.526	6439.821	7772.100
Oct 2014	7352.476	6902.050	7802.903	6663.609	8041.344
Nov 2014	7588.955	7130.822	8047.087	6888.302	8289.607
Dec 2014	7750.958	7268.747	8233.169	7013.480	8488.436

Figure 28 – 12-month forecast according to ARIMA fit.

Part III: Holt's exponential smoothing & results – Exponential smoothing models are able to fit various types of components in time series [14]. Single exponential smoothing "fits a time series that has level and irregularity", while double and triple smoothing add trend and seasonality (respectively) [14]. In figure 14 above, we decomposed our time series data into these three additive components. The Holt approach is double exponential smoothing, and "can fit a time series that has an overall level and a trend" [15]. The Holt-Winters is triple smoothing, which tracks the seasonable component. For this case, two versions of Holt's approach were compared to find a better fit: one which is additive and the other multiplicative.

Both types of models are illustrated in the figures below. As seen in figure 33, Holt's multiplicative model has lower & better error statistics than the additive model. In observing the resultant forecasts, figure 30 indicates that the multiplicative forecast has a nonlinear trend and larger range of quantiles. Upon reviewing the Q-Q plot of residual distribution along a normal distribution (see R project code), the multiplicative model appears to be much more normally distributed. These observations suggest that the multiplicative model more accurately fits the data and will serve as a better forecast than the additive model.

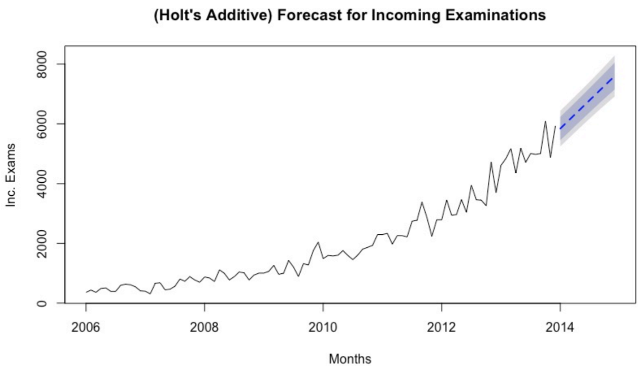


Figure 29 – Plot of Holt's additive forecast

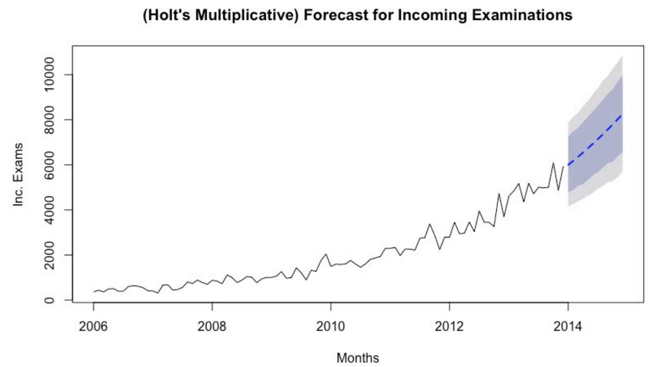


Figure 30 – Plot of Holt's multiplicative forecast

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2014	5841.241	5450.788	6231.695	5244.094	6438.389
Feb 2014	6001.130	5610.291	6391.968	5403.394	6598.866
Mar 2014	6161.018	5769.316	6552.721	5561.961	6760.076
Apr 2014	6320.907	5927.673	6714.141	5719.507	6922.306
May 2014	6480.795	6085.181	6876.410	5875.755	7085.835
Jun 2014	6640.684	6241.666	7039.701	6030.439	7250.928
Jul 2014	6800.572	6396.969	7204.175	6183.315	7417.829
Aug 2014	6960.460	6550.946	7369.975	6334.162	7586.759
Sep 2014	7120.349	6703.473	7537.225	6482.792	7757.906
Oct 2014	7280.237	6854.448	7706.026	6629.049	7931.425
Nov 2014	7440.126	7003.796	7876.455	6772.817	8107.434
Dec 2014	7600.014	7151.463	8048.565	6914.014	8286.014

Figure 31 – 12-month forecast according to Holt's (additive) smoothing

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2014	5998.765	4784.575	7252.659	4156.633	7854.464
Feb 2014	6175.419	4881.349	7480.792	4258.802	8129.167
Mar 2014	6357.274	5074.995	7642.245	4391.456	8311.894
Apr 2014	6544.485	5143.164	7890.217	4490.479	8602.715
May 2014	6737.209	5339.955	8122.499	4635.501	8829.059
Jun 2014	6935.608	5530.194	8373.552	4752.705	9132.123
Jul 2014	7139.850	5676.106	8616.208	4940.533	9397.457
Aug 2014	7350.106	5876.454	8876.976	5057.013	9736.108
Sep 2014	7566.554	6072.970	9148.543	5236.658	9945.161
Oct 2014	7789.376	6136.851	9355.334	5287.093	10258.338
Nov 2014	8018.760	6380.536	9680.602	5447.042	10553.538
Dec 2014	8254.898	6559.881	9978.902	5717.386	10855.223

Figure 32 – 12-month forecast according to Holt's (multiplicative) smoothing

```
> accuracy(fitholts_m)
                   ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set 9.777434 269.1309 198.3447 -0.629297 12.30327 0.2962633 -0.1297505
>  accuracy(fitholts_a)
                   ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 67.84433 298.2579 208.7351 0.2865199 13.20034 0.3117831 -0.009329275
```

Figure 33 – Comparison of model accuracy between both approaches (additive and multiplicative).

Part IV: ARIMA vs Holt's – Both of the ARIMA models as well as Holt's multiplicative model are worthy forecasts in this analysis. To be precise, however, the ARIMA *custom* model and Holt's *multiplicative* model have lower error statistics, lower residual ranges, lower S^2 (variance), and higher correlation between predicted and observed values as compared to the other two models. At the same time, the auto ARIMA model has higher non-significance in the Ljung-Box test (suggesting lower AC and better chi-square). This may have been due to lower mean error in the autofit. As a result, the custom ARIMA and multiplicative Holt models fit the imputed time series better (more closely). The risk of the custom fit ARIMA model is possible overfitting when applied to a broader set of input data. For this reason, it would be prudent to compare all three models in computing a single forecast.

Summary and Recommendations

The purpose of this project was to develop a "a predictive analytic product which could help Fargo accurately guess incoming volume of medical requests and use that information to improve the scheduling of physicians" [3].

Summarized in another way, the goal was to create a model. In this project, I described my strategy and approach, the stages of cleaning, imputation, and forecasting, and referenced in detail how each stage was assessed and executed. Also, I discussed the data problems, how they were analyzed, cleaned and imputed. We reviewed and implemented two methods of imputation, illustrated the process, and provided a full analysis of comparative results. Proceeding, we also covered two different forecasting methods in detail, produced four different forecasting models, and explained analytical considerations and results. In conclusion, as described in part four of stage three, we offer Fargo Health Group three different forecasting models that can effectively serve as a predictive tool for forecasting examinations and improving scheduling.

With this in mind, a few recommendations are in order. First, Fargo Health Group should remember that the sample it provided as the basis for this pilot study is only one of a larger population. It must test the models that we have generated against other samples and the larger population. Only after developing a model that is applicable to the general population can it confidently depend on its predictive capabilities. Secondly, Fargo needs to remember that a model is only as good as its inputs. If it can collect predictor variables in addition to 'month' and 'year', it would be able to create other types of potentially more accurate models. Thirdly, Fargo should remember that a model is subject to business assumptions and real-world constraints that may not be incorporated as predictive variables. This is to say, Fargo needs to have a better pulse on (socioeconomic) factors that drive examinations in the first place. More research should be done on the customer side of the modeling initiative.

Additionally, we urge Fargo to consider the ethical responsibilities it undertakes in pursuit of predictive technology. As Mr. Rubin of the QAO has stated, "predictive analytics is what the organization needs to reduce" various costs and fees as a result of failing to meet examination demand [2]. It was for this reason that QAO started collecting incoming examination volumes. Fargo needs to make sure that its data collection policies are in line with local and federal regulations and that consumers have offered their consent on these practices where necessary. It is also important that Fargo is transparently using the data that it collects and does not amass more data than it needs. Another obvious but understated ethical consideration is that predictive technology should be used to better the lives of customers instead of only to increase profit margins. The costs and fees that Fargo pays for incomplete examinations were introduced to facilitate better service for customers. Fargo's goals are not only to reduce these fees but to improve its services for customers.

Conclusively, we reiterate that Fargo must exercise equal parts responsibility and ownership in pursuing predictive technology. Predictive analytics offer Fargo a big opportunity to improve its business, but they must be implemented with the priority of putting customer privacy and trust first.

Citations

[1] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 2)**. Shelter Island, NY: Manning.
[2] Khachatryan, D. (2014). Fargo Health Group: Managing the Demand for Medical Examinations Using Predictive Analytics **(p. 3).** Babson College, Wellesley, MA.
[3] Khachatryan, D. (2014). Fargo Health Group: Managing the Demand for Medical Examinations Using Predictive Analytics **(p. 4).** Babson College, Wellesley, MA.
[4] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 427)**. Shelter Island, NY: Manning.
[5] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 428)**. Shelter Island, NY: Manning.
[6] Rdocumentation.org. (2018). pool function | R Documentation. [online] Available at: https://www.rdocumentation.org/packages/mice/versions/3.1.0/topics/pool [Accessed 20 Jul. 2018].
[7] Understanding Diagnostic Plots for Linear Regression Analysis (2018) | University of Virginia Library: Data Services + Sciences. [online] Available at: https://data.library.virginia.edu/diagnostic-plots/ [Accessed 20 Jul. 2018].
[8] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 359)**. Shelter Island, NY: Manning.
[9] ARIMA models for time series and forecasting (2018) | Duke University: Statistical forecasting: notes on regression and time series analysis. [online] Available at: https://people.duke.edu/~rnau/411arim.htm [Accessed 20 Jul. 2018].
[10] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 360)**. Shelter Island, NY: Manning.
[11] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 361)**. Shelter Island, NY: Manning.
[12] Stationarity (2018) | Engineering Statistics Handbook. [online] Available at: https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc442.htm [Accessed 20 Jul. 2018].
[13] Stationarity and differencing (2018) | Duke University: Statistical forecasting: notes on regression and time series analysis. [online] Available at: https://people.duke.edu/~rnau/411arim.htm [Accessed 20 Jul. 2018].
[14] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 352)**. Shelter Island, NY: Manning.
[15] Kabacoff, R. (2015). R in action: Data analysis and graphics with R **(p. 355)**. Shelter Island, NY: Manning.

2018 Copyright Eli Bolotin, All Rights Reserved.