Final Project
Michael Pallante

The problem that will be explored is the dengue epidemic and the task at hand is to make accurate predictions of dengue in two selected cities: San Juan, Puerto Rico and Iquitos, Peru. The dengue virus is transmitted by mosquitoes and puts the population of these two tropic cities at risk for illness and even death. The significance of this problem is that accurate dengue predictions for these two cities could potentially reduce the impact of these outbreaks if and when they do occur, which could result in lives saved and less spread of illnesses. To make such predictions, multiple predictive models will be built and submitted to the DengAI: Predicting Disease Spread competition that is hosted by DrivenData. These predictive models will be measured by mean absolute error. The more this metric is minimized, the better the model is at its predictive ability of dengue.

Before the modeling process was done, an exploratory data analysis was performed. Since this data includes two cities, we first filter the data by city and split up the training data set into two different sets, one for each city. Next, we must check to see if the San Juan or Iquitos data sets have missing values that could potentially influence our models. There were a very small number of missing values identified and imputed. To further analyze relationships that exist in the data set, a correlation plot of all variables in each data set was also created. In these plots, we see that some shared variables by both data sets that may have some explanatory power over the data. Particularly, reanalysis_specific_humidity_g_per_kg and reanalysis_dew_point_temp_k are both highly correlated with both of our data sets.

Based on the nature of the data and the need for making future predictions of the data, it was determined that creating a time series out of the data would be the best way to create models for the data. To conclude the exploratory data analysis, the San Juan and Iquitos data sets were both made into a time series. The time series plots of both cities can be seen in Figure 1.1 and Figure 1.2 below. We see that in each of these time series plots, the data shows seasonality with no dominating trends.

Figure 1.1: San Juan Time Series Plot

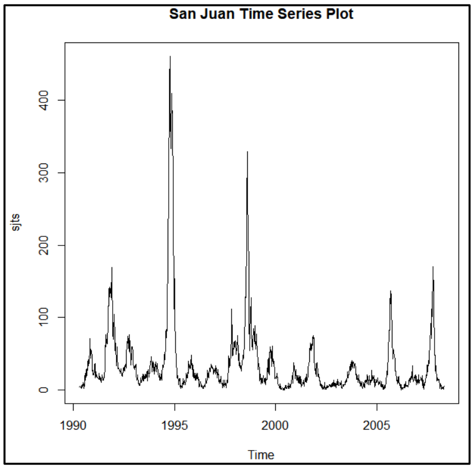
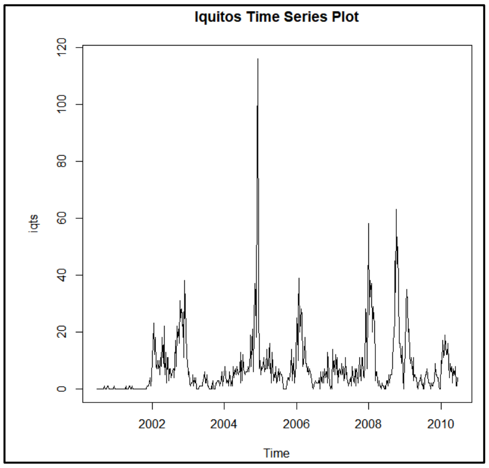


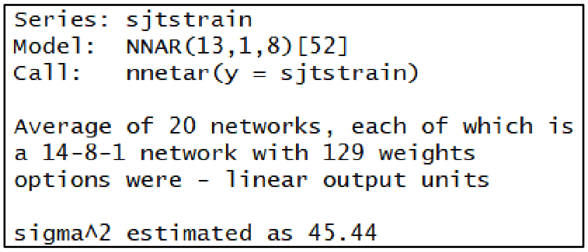
Figure 1.2: Iquitos Time Series Plot



To improve model accuracy, the San Juan and Iquitos time series were each split into their own training and test data sets. Those splits are approximately 80-20 splits, meaning the training sets account for approximately 80% of the data and the test sets account for approximately 20% of the data. The upcoming models and model forecasts will be created on the training data and then tested against the test data for accuracy. This methodology will be performed on all models created. It was determined that three different models will be built for each city, meaning a total of six models. Each model forecast will match the length of the provided submission format document, which contains a total of 260 observations. So, our model forecast predictions will project 260 future dengue cases, for each city.

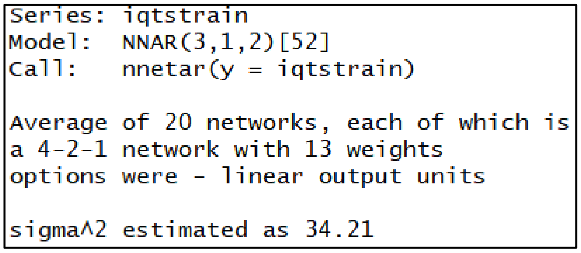
The first model type that was built was a neural network model. This was achieved in R using the nnetar command, which automatically selects the neural network model. The neural network model that was selected for San Juan was the NNAR(13,1,8)[52] model. The neural network model that was selected for Iquitos was the NNAR(3,1,2)[52] model. The formulation of these models can be seen in Figure 2.1 and Figure 2.2 below. The forecasts for these models can be seen in Figure 2.3 and Figure 2.4 below.

Figure 2.1: San Juan NNET Model Formulation



```
Series: sjtstrain
Model:  NNAR(13,1,8)[52]
Call:   nnetar(y = sjtstrain)

Average of 20 networks, each of which is
a 14-8-1 network with 129 weights
options were - linear output units

sigma^2 estimated as 45.44
```

Figure 2.2: Iquitos NNET Model Formulation



```
Series: iqtstrain
Model:  NNAR(3,1,2)[52]
Call:   nnetar(y = iqtstrain)

Average of 20 networks, each of which is
a 4-2-1 network with 13 weights
options were - linear output units

sigma^2 estimated as 34.21
```

Figure 2.3: San Juan NNET Model Forecast

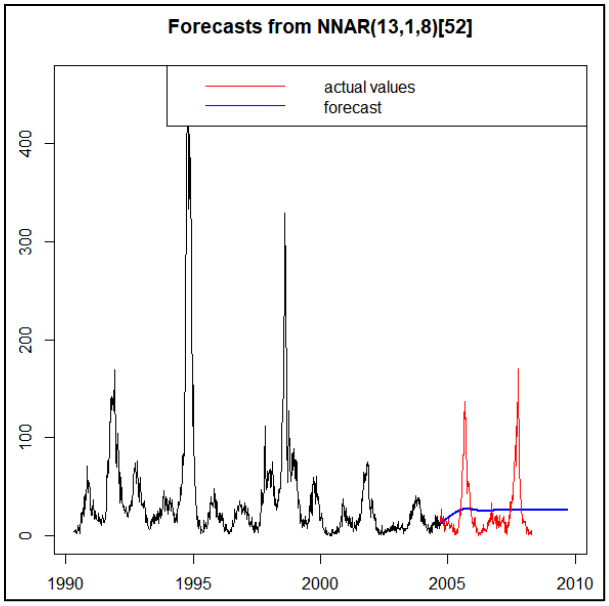
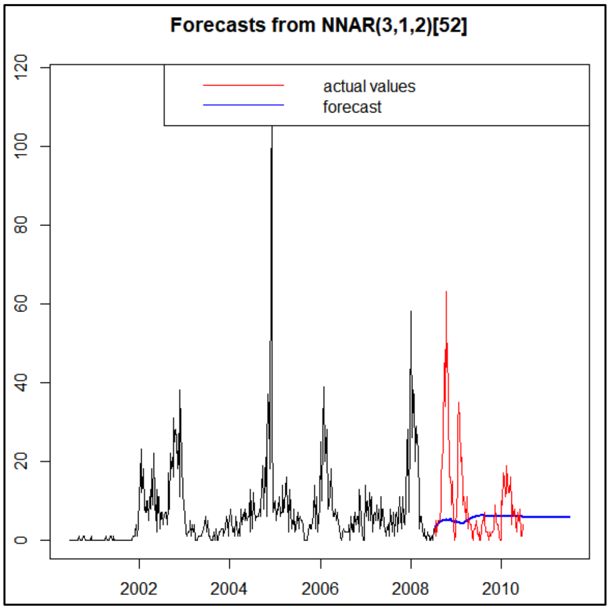
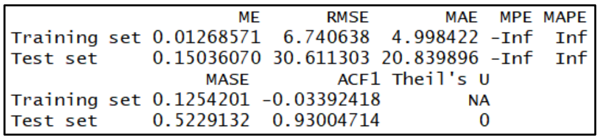


Figure 2.4: Iquitos NNET Model Forecast



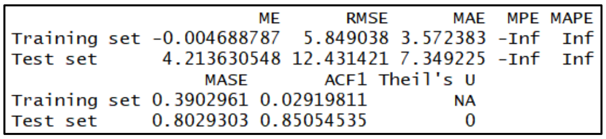
The model accuracy of both San Juan and Iquitos neural network models in comparison to the test data set can be seen in Figure 2.5 and Figure 2.6 below.

Figure 2.5: San Juan NNET Model Accuracy



```
                     ME      RMSE       MAE  MPE MAPE
Training set 0.01268571  6.740638  4.998422 -Inf  Inf
Test set     0.15036070 30.611303 20.839896 -Inf  Inf
                  MASE        ACF1 Theil's U
Training set 0.1254201 -0.03392418        NA
Test set     0.5229132  0.93004714         0
```

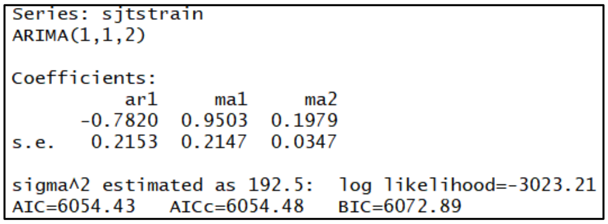
Figure 2.6: Iquitos NNET Model Accuracy



```
                      ME      RMSE      MAE  MPE MAPE
Training set -0.004688787  5.849038 3.572383 -Inf  Inf
Test set      4.213630548 12.431421 7.349225 -Inf  Inf
                  MASE       ACF1 Theil's U
Training set 0.3902961 0.02919811        NA
Test set     0.8029303 0.85054535         0
```

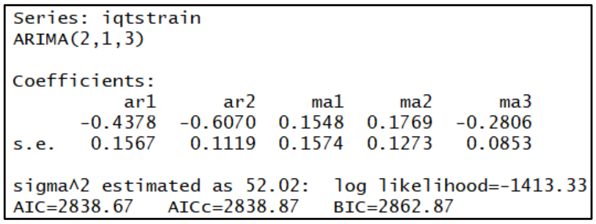
The second model type that was built was an ARIMA model. This was achieved in R using the auto.arima command, which automatically selects the ARIMA model. The ARIMA model that was selected for San Juan was the ARIMA(1,1,2) model. The ARIMA model that was selected for Iquitos was the ARIMA(2,1,3) model. The formulation of these models can be seen in Figure 3.1 and Figure 3.2 below. The forecasts for these models can be seen in Figure 3.3 and Figure 3.4 below.

Figure 3.1: San Juan ARIMA Model Formulation



```
Series: sjtstrain
ARIMA(1,1,2)

Coefficients:
          ar1     ma1     ma2
      -0.7820  0.9503  0.1979
s.e.   0.2153  0.2147  0.0347

sigma^2 estimated as 192.5:  log likelihood=-3023.21
AIC=6054.43   AICc=6054.48   BIC=6072.89
```

Figure 3.2: Iquitos ARIMA Model Formulation



```
Series: iqtstrain
ARIMA(2,1,3)

Coefficients:
          ar1      ar2     ma1     ma2      ma3
      -0.4378  -0.6070  0.1548  0.1769  -0.2806
s.e.   0.1567   0.1119  0.1574  0.1273   0.0853

sigma^2 estimated as 52.02:  log likelihood=-1413.33
AIC=2838.67   AICc=2838.87   BIC=2862.87
```

Figure 3.3: San Juan ARIMA Model Forecast

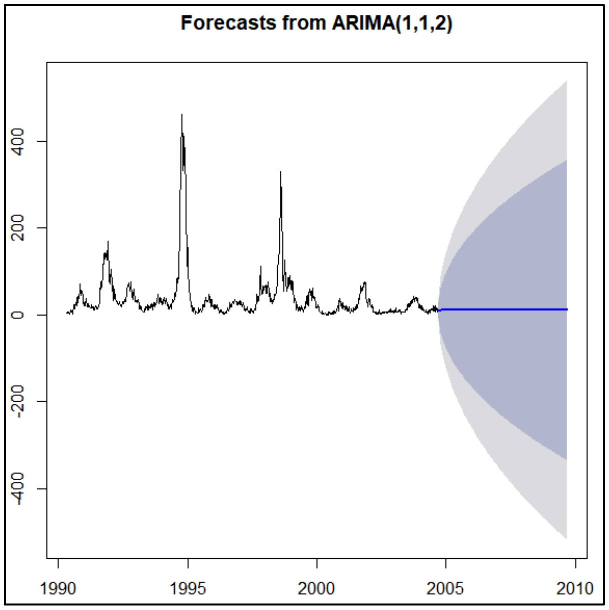
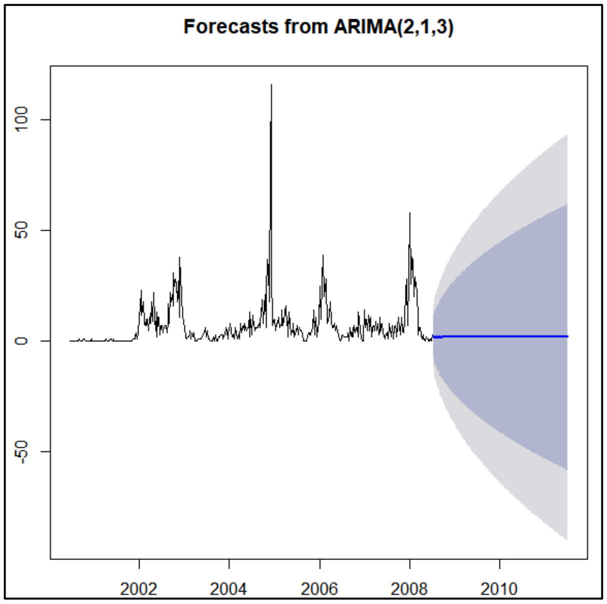
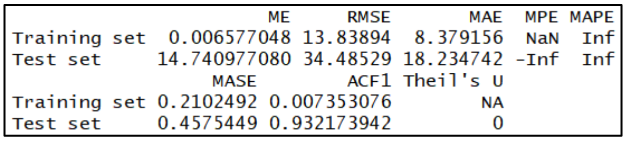


Figure 3.4: Iquitos ARIMA Model Forecast



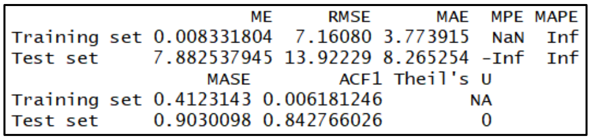
The model accuracy of both San Juan and Iquitos ARIMA models in comparison to the test data set can be seen in Figure 3.5 and Figure 3.6 below.

Figure 3.5: San Juan ARIMA Model Accuracy



```
                      ME     RMSE       MAE  MPE MAPE
Training set  0.006577048 13.83894  8.379156  NaN  Inf
Test set     14.740977080 34.48529 18.234742 -Inf  Inf
                  MASE        ACF1 Theil's U
Training set 0.2102492 0.007353076        NA
Test set     0.4575449 0.932173942         0
```

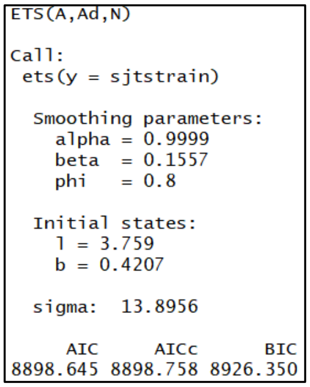
Figure 3.6: Iquitos ARIMA Model Accuracy



```
                      ME     RMSE      MAE  MPE MAPE
Training set 0.008331804  7.16080 3.773915  NaN  Inf
Test set     7.882537945 13.92229 8.265254 -Inf  Inf
                  MASE        ACF1 Theil's U
Training set 0.4123143 0.006181246        NA
Test set     0.9030098 0.842766026         0
```

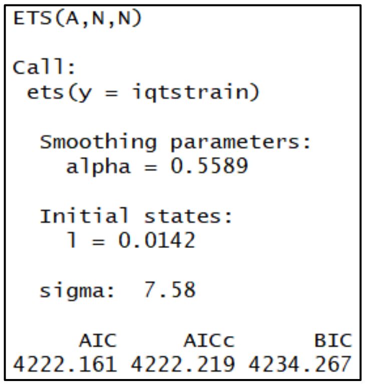
The third model type that was built was an ETS model. This was achieved in R using the ets command, which automatically selects the ETS model. The ETS model that was selected for San Juan was the ETS(A,Ad,N) model. The ETS model that was selected for Iquitos was the ETS(A,N,N) model. The formulation of these models can be seen in Figure 4.1 and Figure 4.2 below. The forecasts for these models can be seen in Figure 4.3 and Figure 4.4 below.

Figure 4.1: San Juan ETS Model Formulation



```
ETS(A,Ad,N)

Call:
 ets(y = sjtstrain)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.1557
    phi   = 0.8

  Initial states:
    l = 3.759
    b = 0.4207

  sigma:  13.8956

     AIC     AICc      BIC
8898.645 8898.758 8926.350
```

Figure 4.2: Iquitos ETS Model Formulation



```
ETS(A,N,N)

Call:
 ets(y = iqtstrain)

  Smoothing parameters:
    alpha = 0.5589

  Initial states:
    l = 0.0142

  sigma:  7.58

     AIC     AICc      BIC
4222.161 4222.219 4234.267
```

Figure 4.3: San Juan ETS Model Forecast

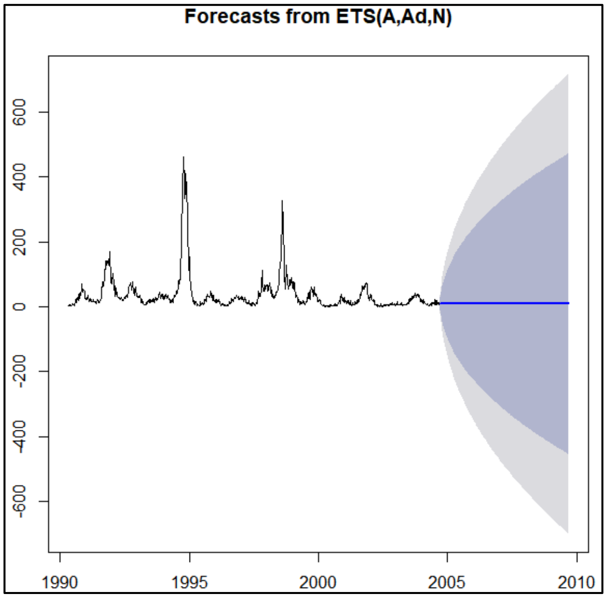
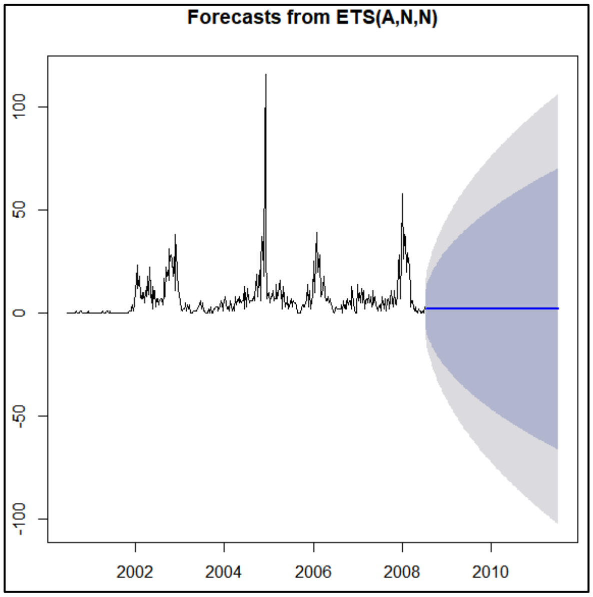
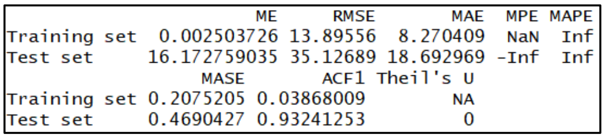


Figure 4.4: Iquitos ETS Model Forecast



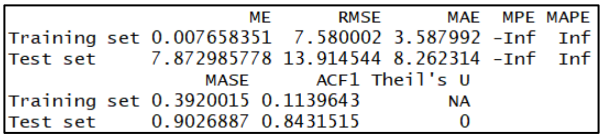
The model accuracy of both San Juan and Iquitos ETS models in comparison to the test data set can be seen in Figure 4.5 and Figure 4.6 below.

Figure 4.5: San Juan ETS Model Accuracy



```
                        ME     RMSE       MAE  MPE MAPE
Training set  0.002503726 13.89556  8.270409  NaN  Inf
Test set     16.172759035 35.12689 18.692969 -Inf  Inf
                  MASE       ACF1 Theil's U
Training set 0.2075205 0.03868009        NA
Test set     0.4690427 0.93241253         0
```

Figure 4.6: Iquitos ETS Model Accuracy



```
                      ME      RMSE      MAE  MPE MAPE
Training set 0.007658351  7.580002 3.587992 -Inf  Inf
Test set     7.872985778 13.914544 8.262314 -Inf  Inf
                  MASE      ACF1 Theil's U
Training set 0.3920015 0.1139643        NA
Test set     0.9026887 0.8431515         0
```

After building all three model types, the predictions from each San Juan and Iquitos model were combined in respective model documents for submission. This allowed for three submissions into the competition: one containing the San Juan and Iquitos neural network model predictions, one containing the San Juan and Iquitos ARIMA model predictions, and one containing the San Juan and Iquitos ETS model predictions. The performance of each model after submission, in terms of our MAE evaluation metric, can be seen in Figure 5.1 below. The lower the MAE, the better the model performed.

Figure 5.1: Model Performance, Post-Submission to DrivenData

Model	**Mean Absolute Error (MAE)**
Neural Network	27.7019
ETS	31.8870
ARIMA	32.8221

The best model submitted to DrivenData for dengue predictions was the neural network model, which had an MAE of 27.7019. At the time of submission, this entry ranked 1088 out of 3506 competitors. The score proved to be a bit below average, with room for improvement. The models designed were likely limited in how they captured the total cases and likely needed greater inclusion of the higher correlated variables to total cases. Perhaps creating models using time series was not the best way to predict the total number of cases, as evidenced by the DrivenData benchmark negative binomial model outperforming all three time series models. In future work, I would have to reconsider using time series to make these predictions, seeing as how there were some high correlations to total cases suggested by my correlation plots. There were multiple variables with shared high correlations in each city. I learned a great deal from this assignment, however, as this was my first competition with DrivenData and only my second-ever competition in general. I learned that traditional machine learning methodology can outperform time series methodology in forecasting. I also expanded my coding abilities again, considering I was a basic R coder with no coding background prior to the MSDS program. I also improved a bit scoring-wise from the midterm competition, suggesting that some growth was achieved.

Literature

This section serves to briefly discuss peer reviewed journals that were used to inspire this report and relate to how this project was completed. The journal titled “Dengue confirmed-cases prediction: A neural network model” documents research that was guided towards predicting dengue cases with the use of neural networks. This model used a larger dataset than ours, however it showed effective modeling techniques using neural networks to forecast dengue outbreaks. The journal titled “Time Series Analysis of Dengue Incidence in Rio de Janeiro, Brazil” demonstrated similarly effective predictive power of dengue cases, but instead used an ARIMA modeling approach to complete this. Their predictive trends were found estimating the number of dengue cases in a month based on previous months’ occurring cases. The journal titled “Forecasting Dengue Haemorrhagic Fever Cases in Southern Thailand using ARIMA Models” also found some success in forecasting monthly number of dengue cases using ARIMA modeling and time series analysis. Their forecasting offered potential for improved contingency plans for dengue incidences. The journal titled “Neural network forecasting for seasonal and trend time series” studies the effectiveness of using neural network models for the modeling of time series data. The study yielded mixed results, however, and gave me reason to examine this data set using ARIMA and ETS models due to differing preprocessing approaches. The journal titled “Deep Learning for Time-Series Analysis” compared the neural network model approach to other modeling alternatives, including ARIMA, and concluded that it produced better results than shallower techniques.

References

Aburas, H. M., Cetiner, B., & Sari, M. (2010). Dengue confirmed-cases prediction: A neural network model. *Expert Systems with Applications, 37*(6), 4256-4260. Retrieved June 3, 2018, from https://www.sciencedirect.com/science/article/pii/S0957417409010197

Gamboa, J. (2017). Deep Learning for Time-Series Analysis. Retrieved June 3, 2018, from https://arxiv.org/abs/1701.01887

Luz, P. M., Mendes, B. V., Codeco, C. T., Struchiner, C. J., & Galvani, A. P. (2008). Time Series Analysis of Dengue Incidence in Rio de Janeiro, Brazil. *The American Journal of Tropical Medicine and Hygiene, 79*(6), 933-939. Retrieved June 3, 2018, from www.ajtmh.org/docserver/fulltext/14761645/79/6/0790933.pdf?expires=1528769161&id=id&accname=guest&checksum=64C71B0CAB281E415928847F7AA888E5

Promprou, S., Jaroensutasinee, M., & Jaroensutasinee, K. (2006). Forecasting Dengue Haemorrhagic Fever Cases in Southern Thailand using ARIMA Models. *Dengue Bulletin, 30*, 99-106. Retrieved June 3, 2018, from http://apps.who.int/iris/bitstream/handle/10665/170355/db2006v30p99.pdf;jsessionid=2BA9444856C543A962FFBB086AF88B16?sequence=1

Zhang, G., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research, 160*(2), 501-514. Retrieved June 3, 2018, from https://www.sciencedirect.com/science/article/abs/pii/S0377221703005484