Alice Tang
Dr. Fulton
ADEC 7460
18 December 2022

Final Paper: Predicting Disease Spread DengAI Competition

I. Introduction

For our final assignment, we were given environmental data collected by US Federal Government Agencies. The data included environmental variables surrounding changes in temperature, precipitation, and vegetation. With this data, we were tasked with predicting the number of Dengue fever each week for two cities— specifically for the cities of San Juan, Puerto Rico and Iquitos, Peru. Exploring this problem provides incredibly valuable knowledge and offers immense benefits to communities as a whole. In particular, having accurate predictions of disease cases would result in lowering transmission rates and reducing the amount of fatalities.

Knowing the number of cases can aid with controlling outbreaks and reducing the transmission of Dengue fever. This knowledge inherently leads to taking better preventive action to mitigate the risk of being infected. In turn, this can increase the spread of awareness which cotntributes to early detection and earlier treatment— heavily improving the chances of survival. Preparing for the incoming number of cases creates more research incentive as well. This is especially important due to the amount of time, money, and effort research requires. Better research elicits more effective treatment and grants doctors a better understanding of how to successfully treat their patients. The number of human lives that can be saved from this disease by predicting the number of total cases make this an important problem to analyze.

II. Data

Before looking at anything, I noticed the dengue_features_train and dengue_labels_train datasets did not separate the two cities into their own datasets. In order to gain better insight, I decided I needed to filter the data from each city respectively in their own datasets.

A: Examining the Data of Each City

Before visualizing the data, I knew how important it was to see the shape— the amount of entries, variables, and missing values. I first took a look at San Juan. As seen below, the San Juan dataset comprises of 24 features, 936 entries, and a significant amount of missing values. This will certainly need to be addressed before going any further, or else there will be gaps in our visualization models.

```
San Juan
     features:  24   entries:  936   labels:  936

                  ndvi_ne                                 ndvi_nw
                      191                                      49
                  ndvi_sw                                 ndvi_se
                       19                                      19
reanalysis_sat_precip_amt_mm                precipitation_amt_mm
                        9                                       9
         station_precip_mm                    station_min_temp_c
                        6                                       6
       station_max_temp_c               station_diur_temp_rng_c
                        6                                       6
       station_avg_temp_c                     reanalysis_tdtr_k
                        6                                       6
reanalysis_specific_humidity_g_per_kg reanalysis_relative_humidity_percent
                        6                                       6
reanalysis_precip_amt_kg_per_m2           reanalysis_min_air_temp_k
                        6                                       6
reanalysis_max_air_temp_k                 reanalysis_dew_point_temp_k
                        6                                       6
    reanalysis_avg_temp_k                     reanalysis_air_temp_k
                        6                                       6
          week_start_date                              weekofyear
                        0                                       0
                     year                                    city
                        0                                       0
```

Next, I took a glance at the Iquitos dataset as seen below. Similar to the San Juan dataset, it consists of 24 features and a significant number of missing values. However, it should be noted that there are only 520 entries— less than San Juan.

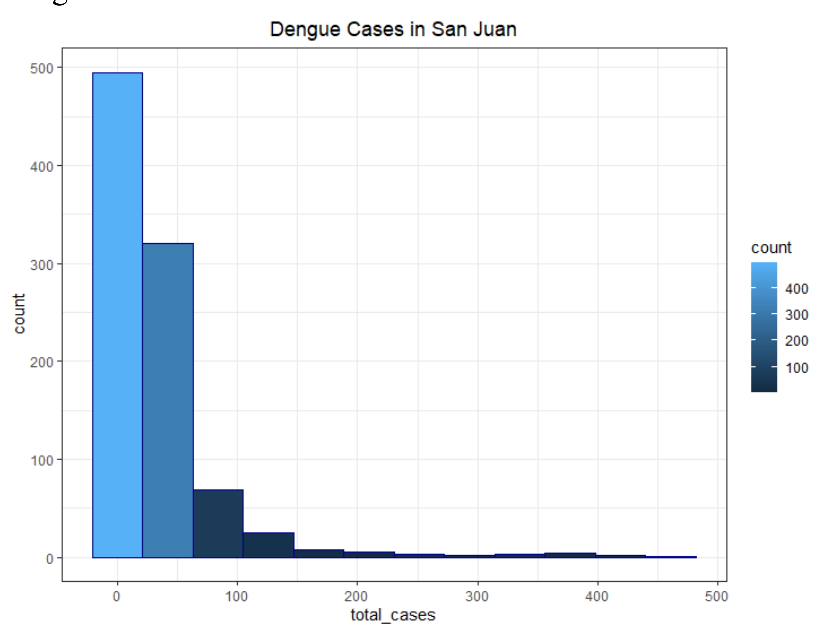
```
Iquitos
     features:  24   entries:  520   labels:  520

                station_diur_temp_rng_c              station_avg_temp_c
                                     37                              37
                       station_precip_mm              station_max_temp_c
                                     16                              14
                      station_min_temp_c               reanalysis_tdtr_k
                                      8                               4
  reanalysis_specific_humidity_g_per_kg      reanalysis_sat_precip_amt_mm
                                      4                               4
   reanalysis_relative_humidity_percent   reanalysis_precip_amt_kg_per_m2
                                      4                               4
               reanalysis_min_air_temp_k        reanalysis_max_air_temp_k
                                      4                               4
             reanalysis_dew_point_temp_k            reanalysis_avg_temp_k
                                      4                               4
                   reanalysis_air_temp_k             precipitation_amt_mm
                                      4                               4
                                 ndvi_sw                          ndvi_se
                                      3                               3
                                 ndvi_nw                          ndvi_ne
                                      3                               3
                         week_start_date                       weekofyear
                                      0                               0
                                    year                             city
                                      0                               0
```

After examining the shape of each city, we can notice that there are many temperature and climate variables in the data. This was not surprising to me, since the transmission of the disease is based on these factors. However, there are also other non-related variables including time variables such as week_start_date. These variables could be helpful if one wanted to create a model with time based features, but for my purposes I only wanted to look at environmental variables. Subsequently, to handle the missing data, I decided to replace the missing values with the previous value. Rather than using the mean or median, I wanted to try out this method since this method seems to be used a lot when observations occur in a time order (Cox). After following these steps to clean the data, I was ready to begin the visualization process.

B: Data Visualization: Bar Plot

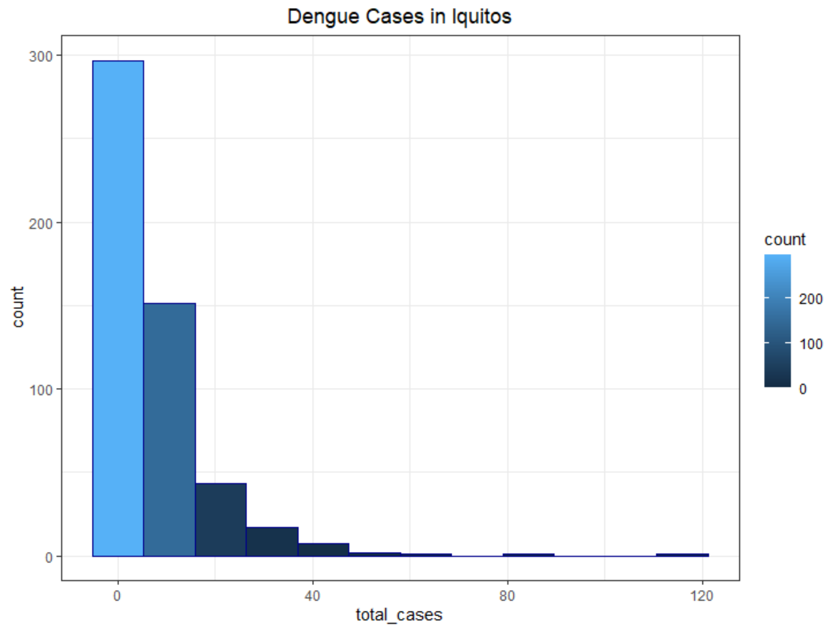
I decided I wanted to see what the bar plots looked like and I also decided to calculate the mean and the variance. First, I looked at the Dengue cases in San Juan.

```
> #SJ
> print(mean(sj_train_labels$total_cases))
[1] 34.18056
> print(var(sj_train_labels$total_cases))
[1] 2640.045
```



As seen in the graphs above, the data is heavily skewed towards the right. This is in accordance to our variance of 2640.05 and our mean of 34.18. The more widespread the data is, the larger the variance is in comparison to the mean. Seeing how spread out our data is, this was not a surprising result. I went through the same process for Iquitos next.

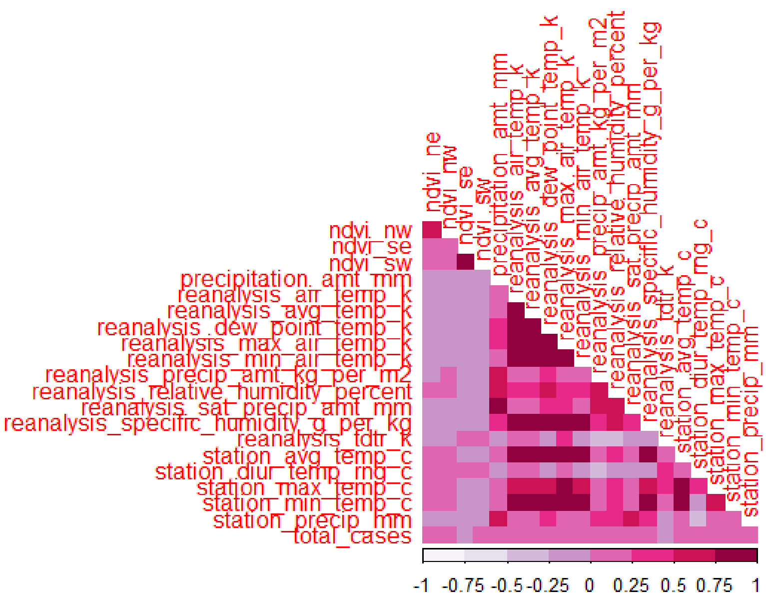
```
> #IQ
> print(mean(iq_train_labels$total_cases))
[1] 7.565385
> print(var(iq_train_labels$total_cases))
[1] 115.8955
```

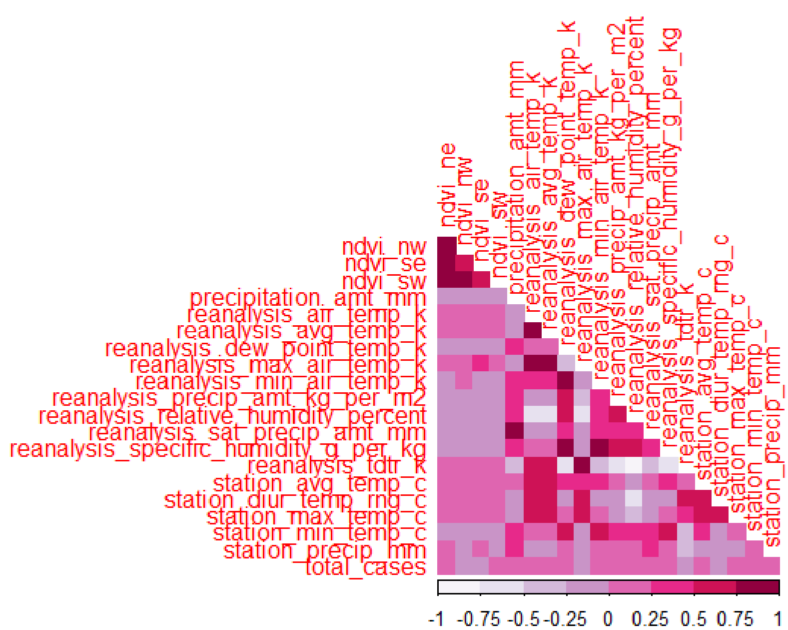


We receive similar results in the graphs above, with a heavy skew towards the right. The Iquitos data has a mean of 7.57 total cases and a 115.9 variance. Though the variance is still higher than the mean, it is a less drastic difference than the San Juan data. This could be due to the fact that there is a smaller timeframe provided for the Iquitos data, thus giving slightly less variation in data.

C: Correlation Plots

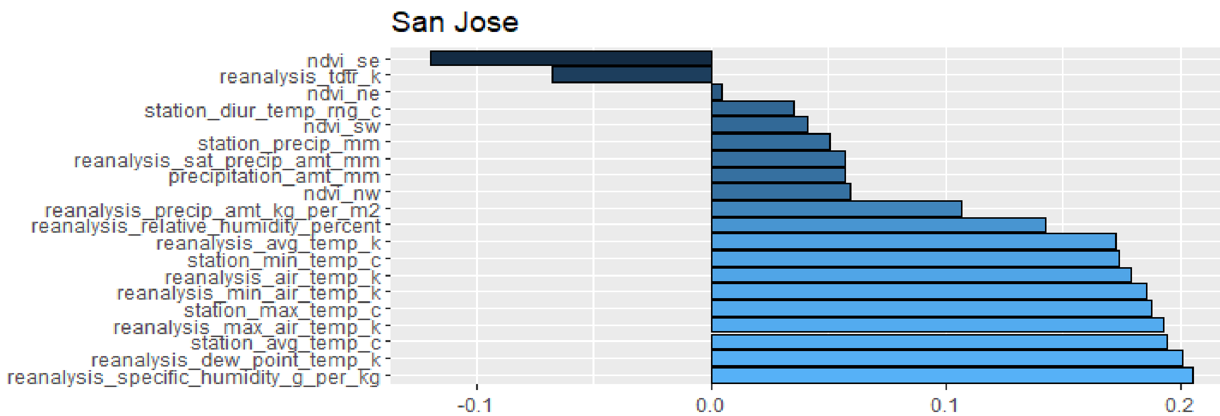
Before moving onto the modeling portion of the competition, I wanted to take a look at the correlation plots of both cities. In turn, I know this will give me a better idea of which variables are significant and highly correlated with another.

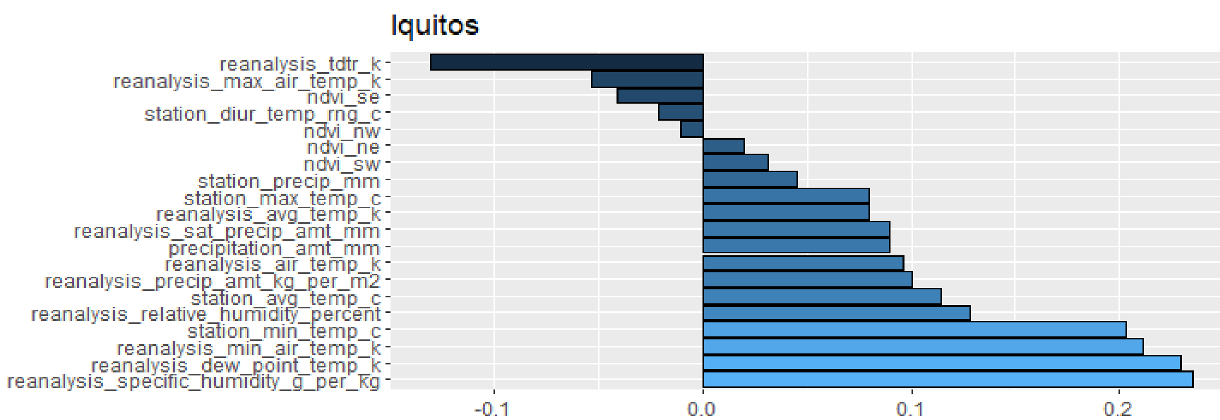




San Juan's correlation plot is located on the left, and Iquitos is on the right. The darker the color is, the higher correlation there is. There are a few takeaways from these correlation plots. First, we can see there is a very strong correlation with all variables relating to temperature and climate. Second, the total_cases variable does not have a notably strong correlation with any variable— only weak correlations. This is the same case for the vegetation variables, there is no strong correlation with any variables. The variables with high correlation to keep in mind include "reanalysis_specific_humidity_g_per_kgm", "reanalysis_dew_point_temp_k", "station_avg_temp_c", and "station_min_temp_c".

Lastly, I wanted to look at the correlation more closely in conjunction with total cases so I could make my final observations before modeling. Below is a correlation bar plot for both cities.





The takeaways for both cities was that with the increase of humidity and wetness, the cases rose up. This makes sense because Dengue fever is transmitted through mosquitos, and mosquitos thrive in these types of environments. We can also see that surprisingly, the amount of precipitation does not have much correlation with total cases, even if humidity does. Lastly, as temperatures plunge or surge to very high temperatures, the total_cases grow. Overall, these findings are not too shocking and make logical sense.

III. Literature

Forecasting COVID-19 Cases Using Time Series Modeling and Association Rule Mining

In this study, researchers used a combination of ARIMA and ARM techniques to predict the number of COVID-19 cases in order to predict the number of cases— aiding in crisis management. The results showed that this ARIMA model had great potential to accurately predict the number of cases, and could be used as a tool in the future to prepare and manage hospital resources during the pandemic.

Overview and Cross-validation of COVID-19 Forecasting Univariate Models

This study tests and uses ETS models to forecast COVID-19 cases. Because ETS models do not require stationarity and has the capability to vary in different models, it was the right decision to use with world data. The study found that of all models tested and used, the ETS model obtained the best results with the lowest MAPE and did not give biased estimates.

Disease Prediction with Different Types of Neural Network Classifiers

This paper explains the valuable contribution of AI and machine learning techniques toward disease prediction. The researchers affirm how neural networks have been successfully used in a variety of scenarios in the medical field. The study states that neural networks are able to improve the "generalization ability of learning systems through training a finite number of neural networks and then combining their results" (Weng et al.).

Prediction of Heart Disease using Multiple Linear Regression Model

In this study, researchers use multiple linear regression to predict heart disease. They found that multiple linear regression was a good model to use, since it is able to explain the relationship between one dependent variable and multiple independent variables. After experimentation, the paper concludes that multiple linear regression was appropriate for predicting heart disease.

A comparative study of SIR Model, Linear Regression, Logistic Function and ARIMA Model for forecasting COVID-19 cases

Researchers tested how well the SIR model, linear regression, logistic function, and the ARIMA model performed for forecasting COVID-19 cases. The study found that the ARIMA outperformed all other models for predictions. The results affirmed that the ARIMA model was able to capture changes in all stages accurately, and had the lowest minimum error.

IV. Types of Models

Ultimately, there were four types of model I knew that I wanted to use. The first two models were the standard ARIMA and ETS models. We specifically talked about and implemented these two models many times in the course, and I came to see them as reliable models. I wanted see how they would perform on this data.

Next, I wanted to try using a Neural Network (NN) model again. Previously when using this model, the performance heavily deteriorated on unseen data. I was adamant to try it again, because NN models seem to be heavily used in disease predictive forecasting.

Lastly, I wished to try the most basic model and common method to predictive analytics that I have learned from introduction courses— a linear regression model. I was curious to see how it would perform because of the correlations we saw previously in the plot. Perhaps some of the relationships were linear and would perform well. I did not have much faith in this basic method, but it proved to be an extremely useful addition.

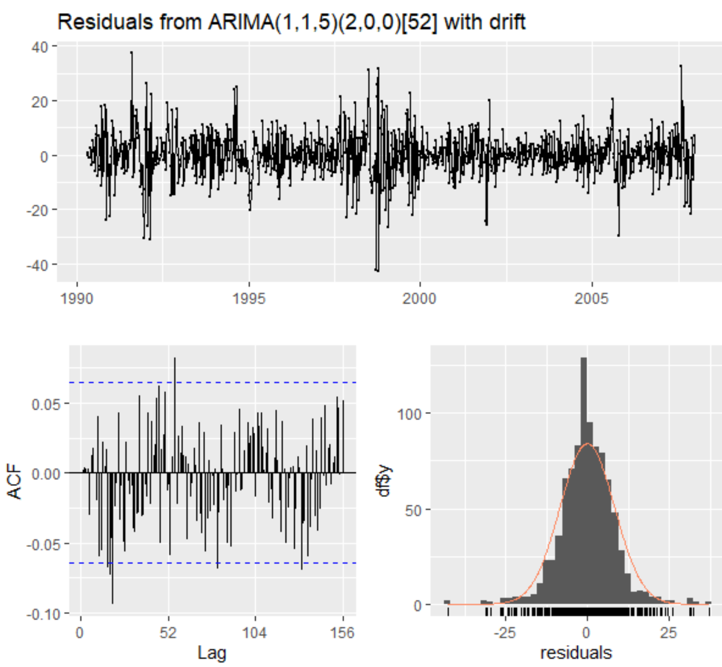
V. Formulation

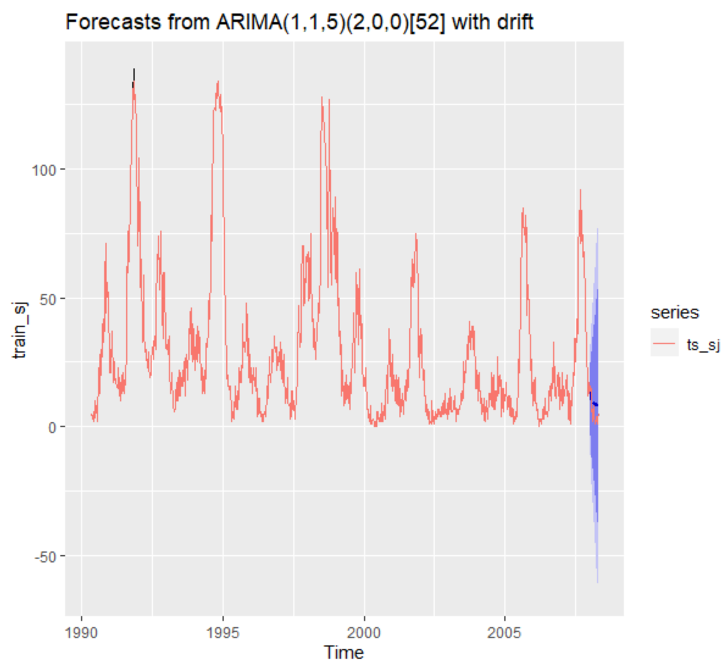
Before beginning the modeling process, I had to convert both the San Juan and Iquitos datasets into time series objects. For San Juan, I chose to only look at predicting 17 weeks since the last full year that was given was 2007. For simplicity, I wanted to only predict the remaining weeks after the last full year of data. Likewise for the Iquitos dataset, I decided to predict the remaining weeks after the last full year of data in 2009. Essentially, these were two different time frames that the problem required you to predict.

A: San Juan Models

1. ARIMA Model

The first model I examined was the ARIMA model. I formulated this model with the auto.arima() function. The model defaulted to ARIMA(1,1,5) with drift. Below are the results.





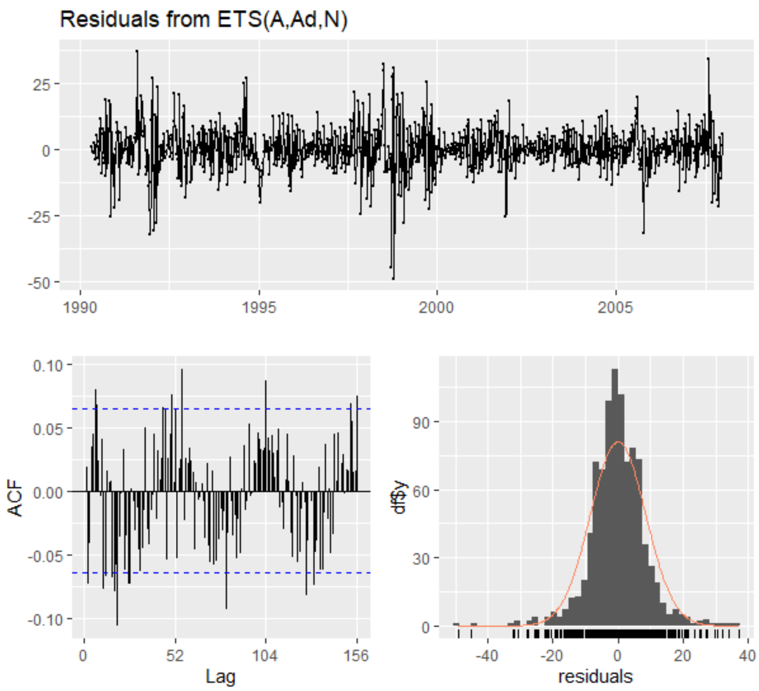

```
> checkresiduals(sj_arima)

        Ljung-Box test

data:  Residuals from ARIMA(1,1,5)(2,0,0)[52] with drift
Q* = 116.17, df = 96, p-value = 0.07897

Model df: 8.   Total lags used: 104
```

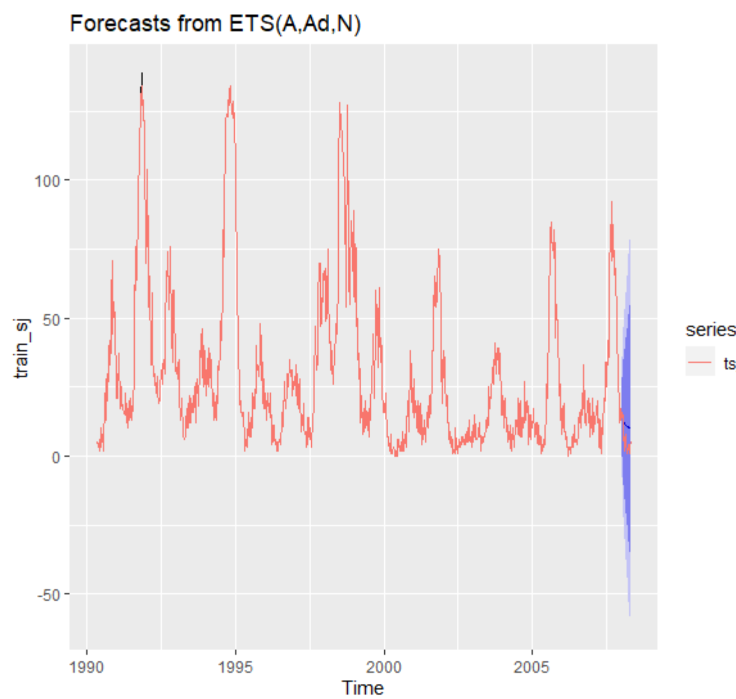
```
Coefficients:
         ar1      ma1     ma2     ma3     ma4     ma5    sar1    sar2   drift
      0.7150  -0.8858  0.0697  0.0383  0.0664  0.0548  0.0665  0.1037  0.0086
s.e.  0.0795   0.0844  0.0471  0.0427  0.0449  0.0370  0.0345  0.0364  0.3906

sigma^2 = 69.82:  log likelihood = -3247.76
AIC=6515.53   AICc=6515.77   BIC=6563.75

Training set error measures:
                       ME     RMSE      MAE  MPE MAPE      MASE        ACF1
Training set -0.002411039 8.310164 6.004817 -Inf  Inf 0.2265341 0.001808867
```

We have obtained a RMSE of 8.31, an MAE of 6, and a AIC of 6515.77. There are a few outliers in the residuals, but in general we still have a normal distribution. Once again, I decided to forecast 17 weeks ahead and the results are in the forecast plot on the right. The forecast seems to capture the essence of the data in the first part, but it does not do as well as the weeks progress.

2. ETS Model

The next model I created was the default ETS model. The optimal ETS model was (A,Ad,N) The results are as shown below.






```
	Ljung-Box test

data:  Residuals from ETS(A,Ad,N)
Q* = 190.07, df = 99, p-value = 1.041e-07

Model df: 5.   Total lags used: 104
```

```
     AIC     AICc      BIC 
10213.55 10213.64 10242.49 

Training set error measures:
                      ME     RMSE      MAE  MPE MAPE      MASE       ACF1
Training set 0.004528378 8.488469 6.053452 -Inf  Inf 0.2283689 0.01907253
```

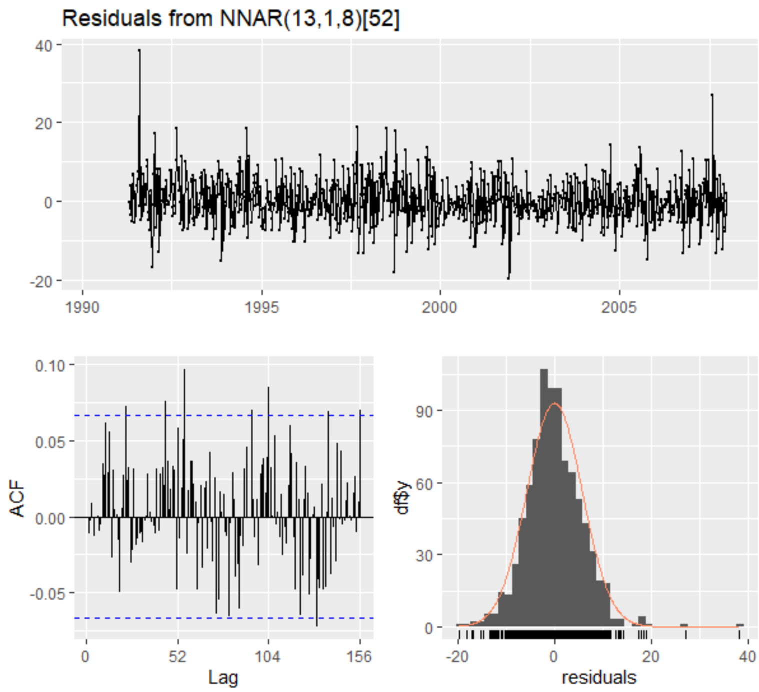
Here, we have obtained an RMSE of 8.49, a MAE of 6.05, and a AIC of 10213.64. Similarly to the ARIMA model, I forecasted 17 weeks into the future to obtain the accuracy. The ETS model seemed to do a little worse than the ARIMA model on the training data. The RMSE and MAE are similar, but the AIC of the ETS model was much higher— signifying a worse performance on the training data.

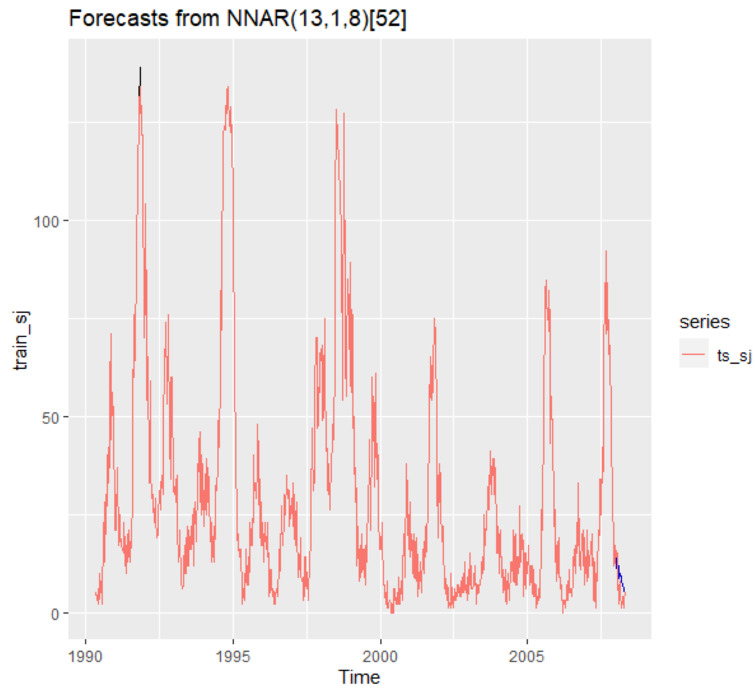
3. Neural Network Model

The third model I used was with the nnetar() function to create a neural network model that forecasted 17 weeks in the future. The results are as follows.


```
	Ljung-Box test

data:  Residuals from ETS(A,Ad,N)
Q* = 190.07, df = 99, p-value = 1.041e-07

Model df: 5.   Total lags used: 104
```


```
          Length Class        Mode     
x         919    ts           numeric  
m           1    -none-       numeric  
p           1    -none-       numeric  
P           1    -none-       numeric  
scalex      2    -none-       list     
size        1    -none-       numeric  
subset    919    -none-       numeric  
model      20    nnetarmodels list     
nnetargs    0    -none-       list     
fitted    919    ts           numeric  
residuals 919    ts           numeric  
lags       14    -none-       numeric  
series      1    -none-       character
method      1    -none-       character
call        2    -none-       call     
```

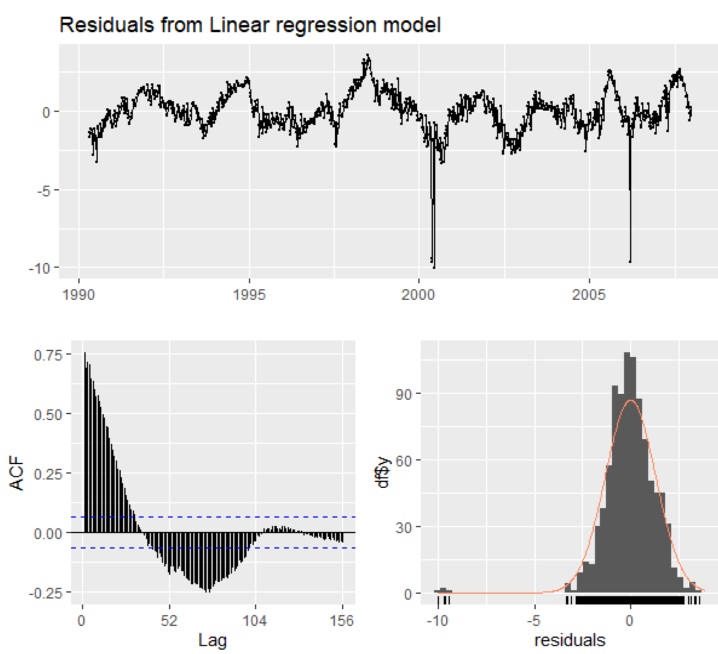


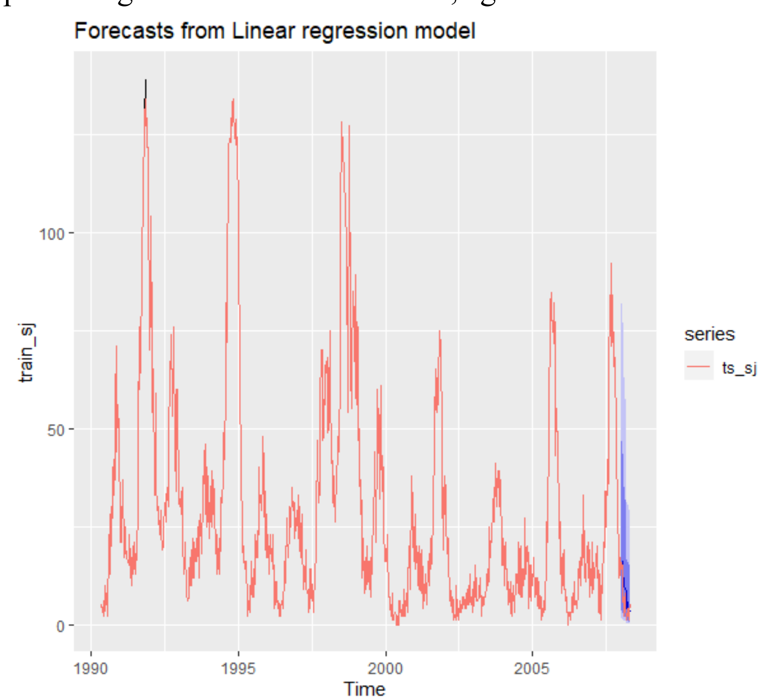


As we can see from the forecasting graph on the right, the NN forecast captures the essence of the training data much better than the ARIMA or ETS model, with a better downward trend. The frequency of [52] was still captured as well.

4. Linear Regression

The last model I used was linear regression predicting 17 weeks into the future, again. The results are below.






```
        Breusch-Godfrey test for serial correlation of order up to 104

data:  Residuals from Linear regression model
LM test = 646.9, df = 104, p-value < 2.2e-16
```

```
Residual standard error: 1.348 on 866 degrees of freedom
Multiple R-squared:  0.3758,    Adjusted R-squared:  0.3383
F-statistic: 10.03 on 52 and 866 DF,  p-value: < 2.2e-16
```

Here, we obtain a very small p-value < 2.2e-16, and the residuals are normally distributed with no outliers. The linear regression forecasting graph looks very promising as well, however, to fully determine the best model, we must conduct an accuracy test.

5. San Juan Model Comparison Results

To evaluate which model had the best results, an accuracy test was performed. The NN model and linear regression model had the best performance of the four models. Ultimately, the linear regression outperformed the NN model slightly with a 3.23 RMSE and 2.65 MAE. However, it is good to note that all of the models' RMSE and MAE were lower on the test set than the train set. Below are the results.

ARIMA

```
> sj.arima.acc
                       ME     RMSE      MAE       MPE    MAPE      MASE        ACF1 Theil's U
Training set -0.002411039 8.310164 6.004817      -Inf     Inf 0.2265341 0.001808867        NA
Test set     -2.767093707 4.522987 3.830157 -160.4722 167.873 0.1444942 0.763798322   3.49692
```

The ARIMA model had a RMSE of 4.52 and a MAE of 3.83 on the testing data.

ETS

```
> sj.ets.acc
                       ME     RMSE      MAE       MPE     MAPE      MASE       ACF1 Theil's U
Training set  0.004528378 8.488469 6.053452      -Inf      Inf 0.2283689 0.01907253        NA
Test set     -5.024838742 6.000875 5.240111 -218.9677 220.4028 0.1976853 0.73509490  4.768506
```

The ETS model had a RMSE of 6 and a MAE of 5.24 on the testing data.

Neural Network

```
> sj.nn.acc
                      ME     RMSE      MAE       MPE     MAPE      MASE        ACF1 Theil's U
Training set -0.02106771 5.722702 4.346775      -Inf      Inf 0.1639838 -0.01136207        NA
Test set     -2.83012397 3.879112 3.369642 -135.4045 139.1522 0.1271211  0.58941465  2.699053
```

The NN model had a RMSE of 3.88 and a MAE of 3.37 on the testing data.

Linear Regression

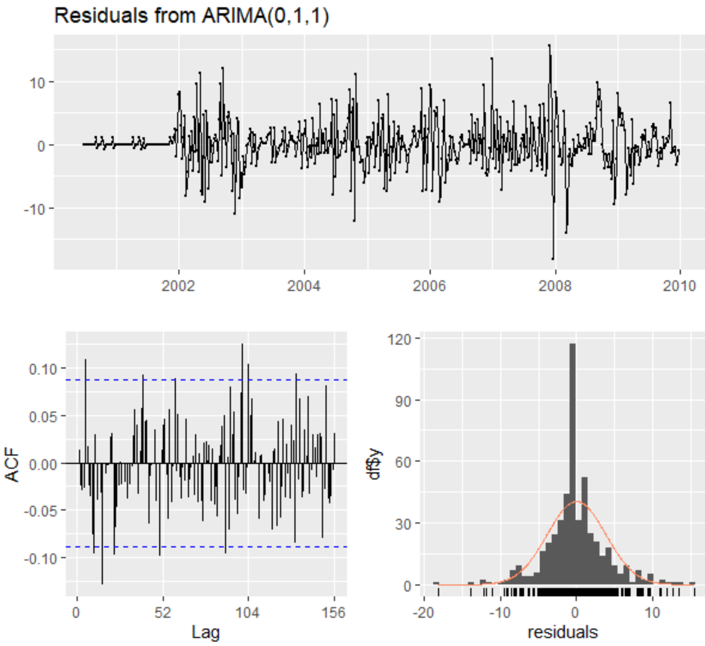
```
> sj.lm.acc
                    ME      RMSE       MAE       MPE     MAPE       MASE      ACF1 Theil's U
Training set  6.220177 24.608929 15.699785      -Inf      Inf 0.59228057 0.9386612        NA
Test set     -2.410425  3.232603  2.646122 -95.96915 100.6831 0.09982599 0.2887530  1.316998
```

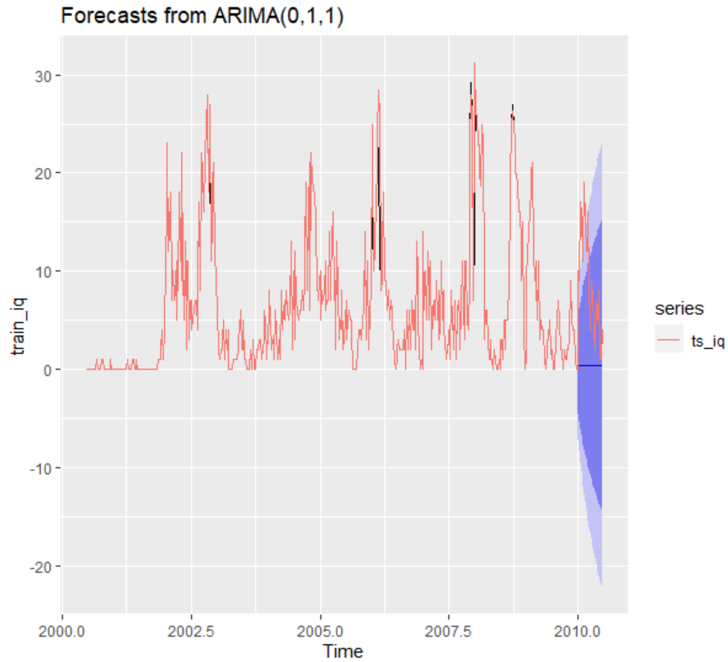
The linear regression model had a RMSE of 3.23 and a MAE of 2.65 on the testing data.

B: Iquitos Models

1. ARIMA Model

Going through the same process as with San Juan, I started off with the ARIMA model again using the auto.arima() function forecasting 25 weeks into the future. The model defaulted to ARIMA(0,1,1). Below are the results.






```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,1)
Q* = 110.37, df = 98, p-value = 0.1852

Model df: 1.   Total lags used: 99
```

```
AIC=2732.88   AICc=2732.91   BIC=2741.29

Training set error measures:
                      ME     RMSE      MAE  MPE MAPE      MASE      ACF1
Training set 0.001394655 3.826243 2.586382 -Inf  Inf 0.3609343 0.013259
```

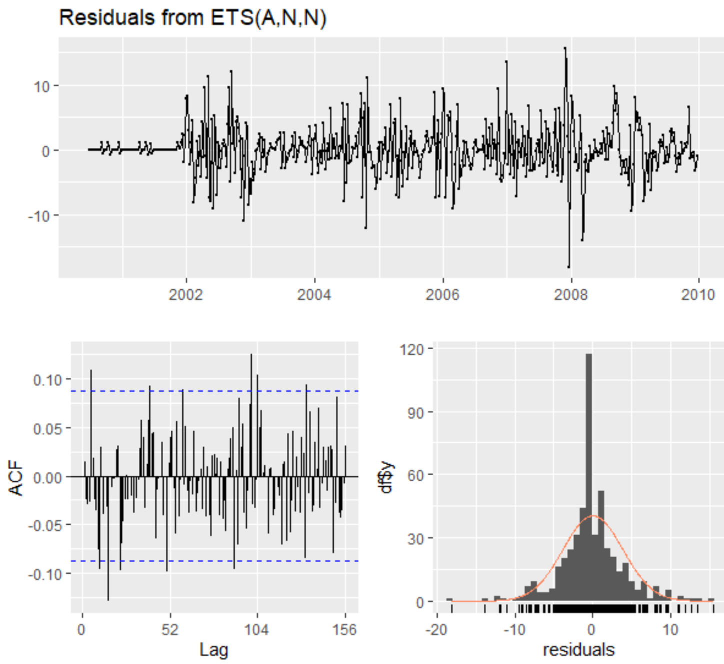
These results look less than optimal, with the forecasting graph being extremely off and not capturing the trend at all. We have a bigger p-value of 0.19, a RMSE of 3.83, and a MAE of 2.59. The residuals are also not fitted under the curve, signifying that the model does not represent the trends in the dataset.

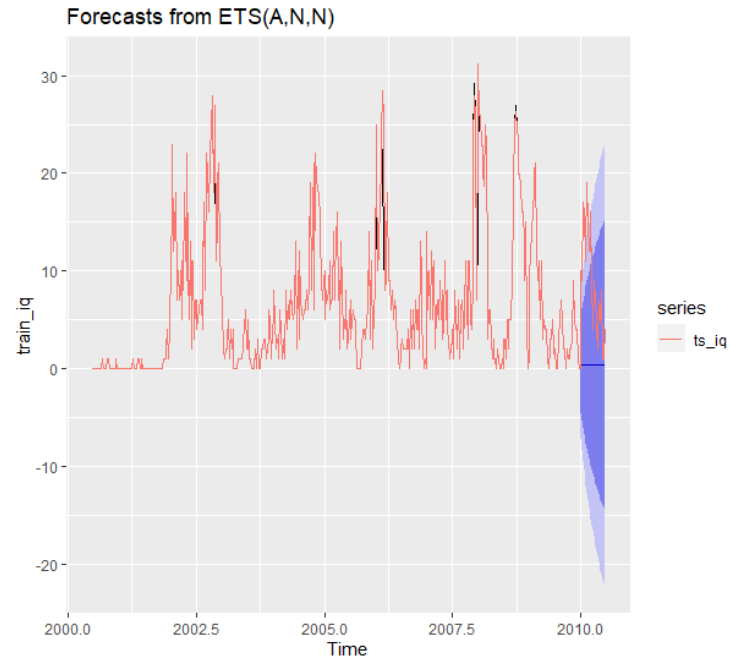
2. ETS Model

The second model was the default ETS model. The optimal ETS model was (A,N,N) The results are as shown below.

```
> checkresiduals(iq_ets)

        Ljung-Box test

data:  Residuals from ETS(A,N,N)
Q* = 110.4, df = 97, p-value = 0.1664

Model df: 2.   Total lags used: 99
```

```
     AIC     AICc      BIC 
4405.720 4405.769 4418.334 

Training set error measures:
                      ME     RMSE      MAE  MPE MAPE      MASE       ACF1
Training set 0.001388173 3.826241 2.586598 -Inf  Inf 0.3609644 0.01421927
```

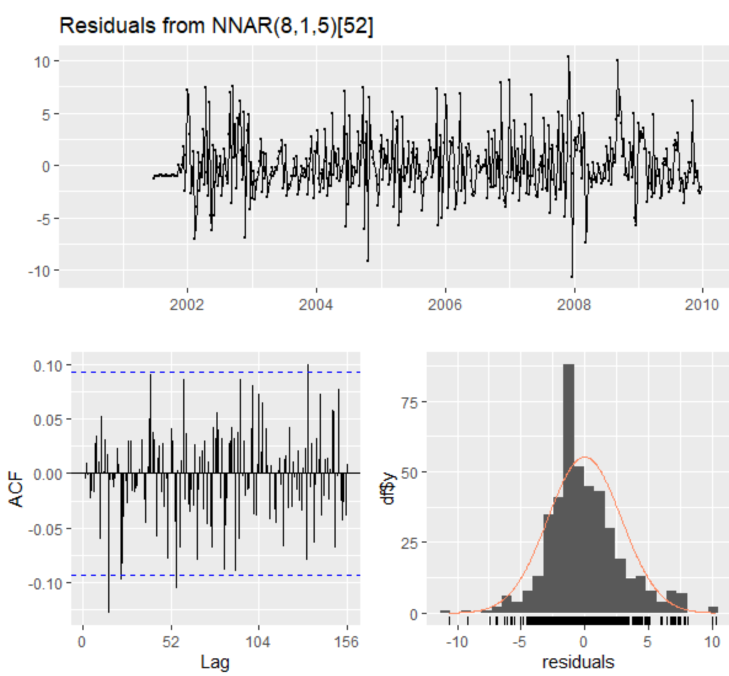


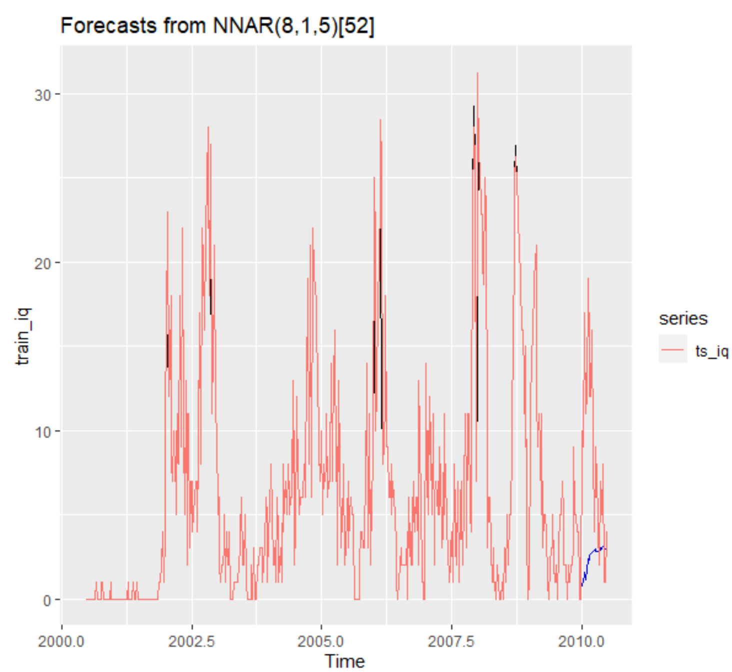


Once again, the trends are not captured well in the data with the forecasting graph being very off and the residuals not fitting under the curve. We see very similar results to the previous ARIMA model with a high p-value of 0.17 and an RMSE of 3.83.

3. Neural Network Model

The third model I used was with the nnetar() function to create a neural network model that forecasted 25 weeks in the future. The results are as follows.



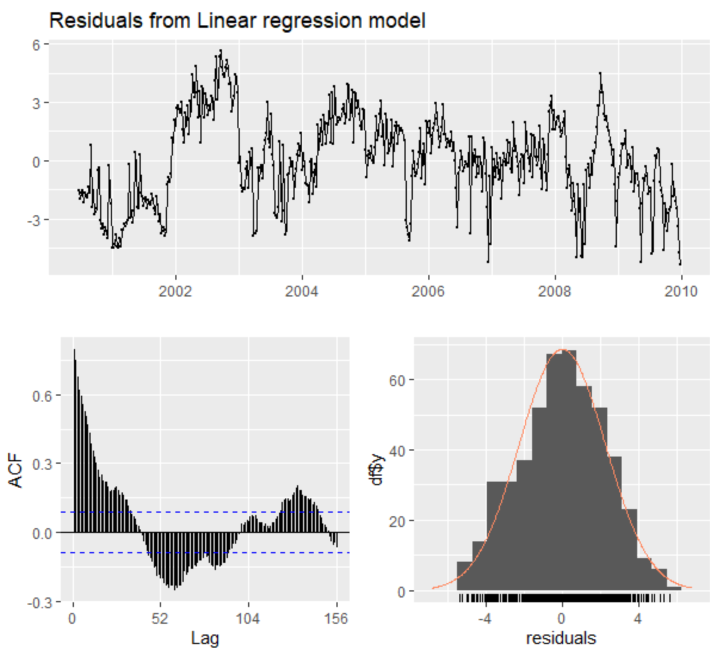


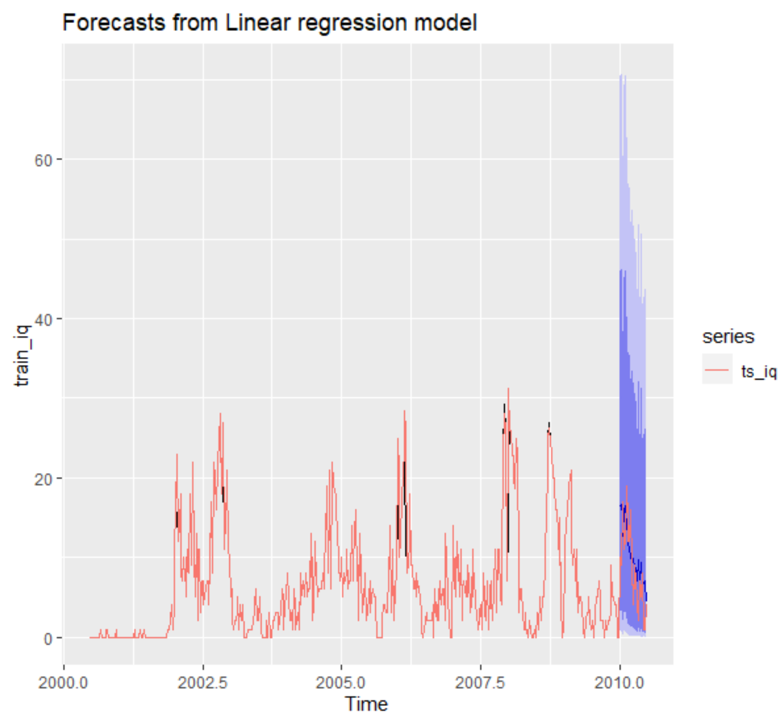
The NN model's performance was also sub-optimal and did not capture the essence of the training data. So far, none of these models have been performing well on the Iquitos data.

```
          Length Class        Mode
x         495    ts           numeric
m           1    -none-       numeric
p           1    -none-       numeric
P           1    -none-       numeric
scalex      2    -none-       list
size        1    -none-       numeric
subset    495    -none-       numeric
model      20    nnetarmodels list
nnetargs    0    -none-       list
fitted    495    ts           numeric
residuals 495    ts           numeric
lags        9    -none-       numeric
series      1    -none-       character
method      1    -none-       character
call        2    -none-       call
```

4. Linear Regression

The last model for our Iquitos data is our linear regression model. Below are the results.






```
> checkresiduals(iq_lm)

        Breusch-Godfrey test for serial correlation of order up to 99

data:  Residuals from Linear regression model
LM test = 363.67, df = 99, p-value < 2.2e-16
```

```
Residual standard error: 2.393 on 442 degrees of freedom
Multiple R-squared:  0.2465,    Adjusted R-squared:  0.1579
F-statistic: 2.781 on 52 and 442 DF,  p-value: 6.933e-09
```

The linear regression model exhibits much better results than the other three models with a low p-value of < 2.2e-16 and a forecast that captures the trend much better. Nevertheless, the accuracy results should still be examined.

5. Iquitos Model Comparison Results

Similarly to the San Juan model comparisons, the NN model and linear regression models still had the best performance. The linear regression model outperformed by far with a RMSE of 4.17 and MAE of 3.32. In this case, the other three models performed better on the training set than the testing data— contrary to the results of the San Juan model comparisons.

ARIMA

```
> iq.arima.acc
                      ME     RMSE      MAE      MPE     MAPE      MASE      ACF1 Theil's U
Training set 0.001394655 3.826243 2.586382     -Inf      Inf 0.3609343 0.0132590        NA
Test set     8.516593158 9.869446 8.516593 91.52272 91.52272 1.1885060 0.6515399  1.522451
```

The ARIMA model had a RMSE of 9.87 and a MAE of 8.52 on the testing data.

ETS

```
> iq.ets.acc
                      ME     RMSE      MAE      MPE     MAPE      MASE       ACF1 Theil's U
Training set 0.001388173 3.826241 2.586598     -Inf      Inf 0.3609644 0.01421927        NA
Test set     8.514033822 9.867237 8.514034 91.46894 91.46894 1.1881488 0.65153995  1.521772
```

The ETS model had a RMSE of 9.87 and a MAE of 8.51 on the testing data.

Neural Network

```
> iq.nn.acc
                    ME     RMSE      MAE      MPE     MAPE      MASE         ACF1 Theil's U
Training set -0.0037236 2.887467 2.182842     -Inf      Inf 0.3046196 -0.005028077        NA
Test set      6.4657298 8.421408 6.859339 41.80989 77.68142 0.9572331  0.701328969  1.074481
```

The ARIMA model had a RMSE of 8.42 and a MAE of 6.86 on the testing data.

Linear Regression

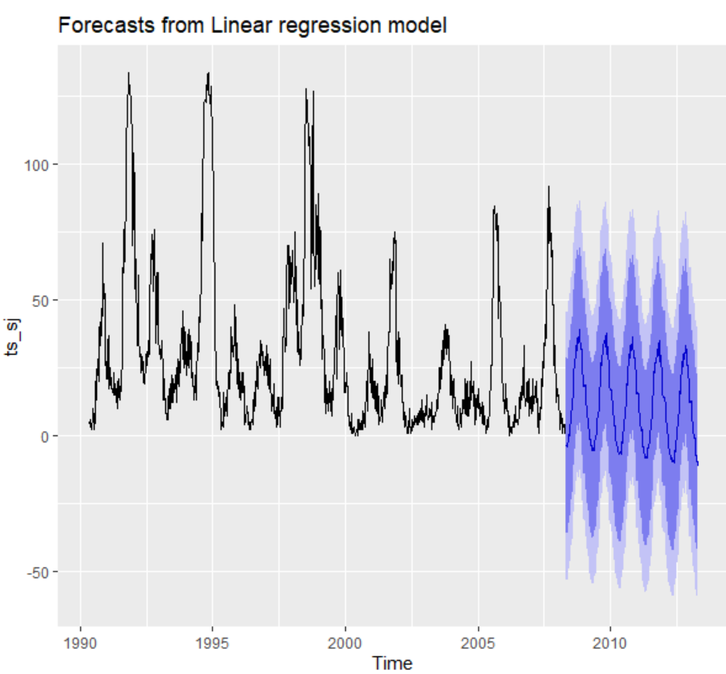
```
> iq.lm.acc
                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1 Theil's U
Training set  2.166852 6.324555 4.256778      -Inf      Inf 0.5940411 0.78880295        NA
Test set     -1.761021 4.165447 3.323195 -83.61744 93.83021 0.4637578 0.01831605  1.469141
```

The linear regression model had a RMSE of 4.17 and a MAE of 3.32 on the testing data.

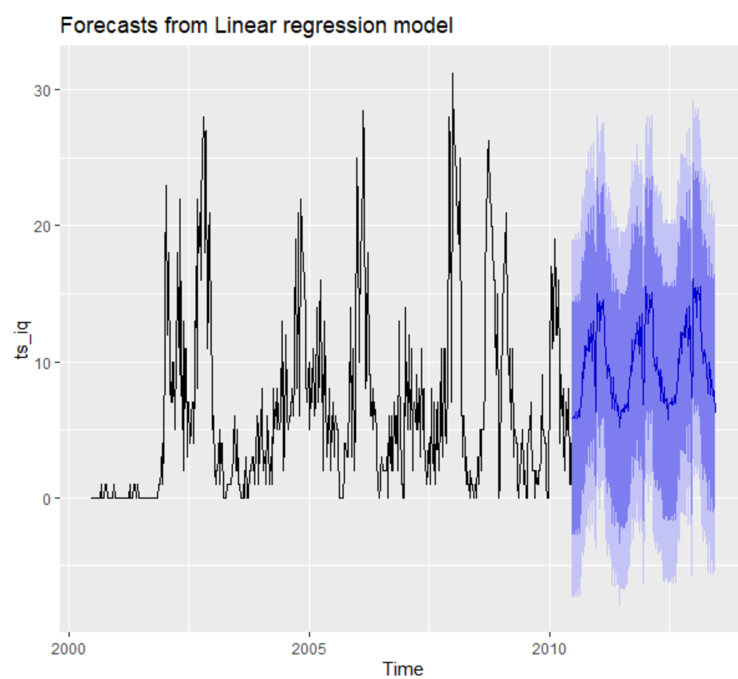
C. Choosing the Final Submission Model for San Juan and Iquitos Data

Regardless of the city, the linear regression model performed best. For the final submission, I used the linear regression model to forecast 260 days for San Jose and 156 days for Iquitos respectively (in accordance to the submission_format). Below are the final forecast graphs that were submitted to the competition.

San Juan



Iquitos



VI. Performance and Accuracy

Score	Submitted by	Timestamp
31.7404	alicetang	2022-12-17 03:44:12 UTC

Overall, my submission scored with a 31.74 MAE. I think that this was an okay attempt considering I used quite a basic model with linear regression. Going into it, I really did not expect to be the top scorer so I anticipated these results.

VII: Limitations and Future Work

The most apparent limitations of my model was the inaccuracy of filling in the missing values. There were a significant amount of NAs and I used a very simple technique of using the previous value to fill them in. This contributed to the inaccuracy, and using a more calculated and sophisticated method would have improved the score. If I reattempted this project, I would have modeled the missing values itself and formed an unbiased estimate.

Even though I was mainly focused on the environmental variables, I acknowledge that not including any time-based features in the model was a major limitation. When examining the time series plot, it can be noted that certain time periods had major spikes but as time passes, the spikes seem to be getting smaller. This pattern could be due to a multitude of reasons such as structural changes in policies or posessing more overall knowledge. Additionally, it will be important to examine more categorical variables such as occupation for a better score. Certain occupations may impact the number of times one is exposed to the virus. To improve my score, I would have accounted for more time-based features and considered other categorical variables and/or historical events.

VIII. Conclusion

I enjoyed participating in this competition because of how applicable this was in real-world situations, especially since we are still currently suffering through a world pandemic. If we had known about COVID-19 earlier and foreseen it, I can only imagine how much it would have helped with preparation on how to handle the situation both government-wise and community-wise. As always, I learned that there are so many different choices of forecasting models, and that it is difficult to find the best fit. I was also shocked at how a basic regression model could outperform the other more advanced models— this makes it important not to dismiss any type of model and to consider all options when forecasting.

Works Cited

Abolmaali, Saina, and Samira Shirzaei. “A Comparative Study of SIR Model, Linear Regression, Logistic Function and ARIMA Model for Forecasting COVID-19 Cases.” *AIMS Public Health*, 26 Aug. 2021, https://www.aimspress.com/article/doi/10.3934/publichealth.2021048.

Atchadé, Mintodê Nicodème, and Yves Morel Sokadjo. “Overview and Cross-Validation of COVID-19 Forecasting Univariate Models.” *Alexandria Engineering Journal*, Science Direct, 21 Aug. 2021, https://www.sciencedirect.com/science/article/pii/S1110016821005470.

Cox, Nicholas. “How Can I Replace Missing Values with Previous or Following Nonmissing Values or Within Sequences?” *Stata*, https://www.stata.com/support/faqs/data management/replacing-missing-values/.

K, Polaraju, and Prasad Durga D. “Prediction of Heart Disease Using Multiple Linear Regression Model .” *International Journal of Engineering Development and Research*, 2017, https://ijedr.org/papers/IJEDR1704226.pdf.

Somyanonthanakul, Rachasak, et al. “Forecasting COVID-19 Cases Using Time Series Modeling and Association Rule Mining.” *BioMed Central*, BMC Medical Research Methodology, 1 Nov. 2022, https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-022-01755-x.

Weng, Cheng-Hsiung, et al. “Disease Prediction with Different Types of Neural Network Classifiers.” *Telematics and Informatics*, Science Direct, 21 Aug. 2015, https://www.sciencedirect.com/science/article/pii/S0736585315001057.