Final Project MSDS 413
Dalya Adams
6/2/2018

Problem

The problem under consideration is to predict the total number of future dengue fever cases in both San Juan, Puerto Rico and Iquitos, Peru, each week. We are provided with NOAA climate and precipitation data, PERSIANN precipitation information, a normalized difference in vegetation index, and 3 years of dengue fever cases for San Juan and 5 years of dengue fever cases for Iquitos.

Significance

This problem is significant because the impact of dengue fever in Latin America is significant, with nearly half a billion cases per year. By predicting the number of cases of dengue fever per week in San Juan, Puerto Rico and Iquitos, Peru, and gaining a better understanding of the relationship between climate and dengue fever, research and resource allocation can be better targeted to eliminate or alleviate the impact of this pandemic. As stated by the Center of Disease Control, "accurate dengue predictions would help public health workers and people around the world take steps to reduce the impact of these epidemics."

Data

Prior to engineering any features, it is important to look at the raw data. We notice that the majority of the data points are for San Juan, 936 versus 520 for Iquitos. We also notice that the ndvi variables all have negative values, and there are a number of null values in the dataset that will need to be addressed prior to modeling.

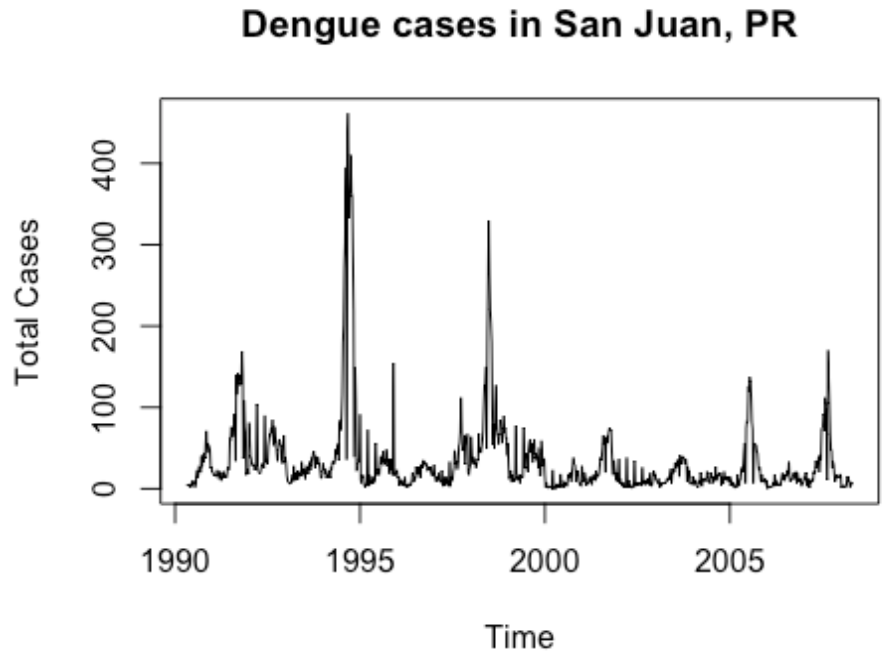
In reference to the ndvi variables, these represent an index of the difference in vegetation. This change in vegetation index leads me to believe that negative values are not errors, and thus they are not adjusted.

```
summary(train)
##  city          year                 weekofyear        total_cases
##  iq:520      Min.   :1990          Min.   : 1.00     Min.   :  0.00
##  sj:936      1st Qu.:1997          1st Qu.:13.75     1st Qu.:  5.00
##              Median :2002          Median :26.50     Median : 12.00
##              Mean   :2001          Mean   :26.50     Mean   : 24.68
##              3rd Qu.:2005          3rd Qu.:39.25     3rd Qu.: 28.00
##              Max.   :2010          Max.   :53.00     Max.   :461.00
##
##   week_start_date              ndvi_ne           ndvi_nw
##  2000-07-01:   2              Min.   :-0.40625  Min.   :-0.45610
##  2000-07-08:   2              1st Qu.: 0.04495  1st Qu.: 0.04922
##  2000-07-15:   2              Median : 0.12882  Median : 0.12143
##  2000-07-22:   2              Mean   : 0.14229  Mean   : 0.13055
##  2000-07-29:   2              3rd Qu.: 0.24848  3rd Qu.: 0.21660
##  2000-08-05:   2              Max.   : 0.50836  Max.   : 0.45443
##  (Other)   :1444              NA's   :194       NA's   :52
```

```
##    ndvi_se                      ndvi_sw                 precipitation_amt_mm
## Min.   :-0.01553   Min.   :-0.06346   Min.   :  0.00
## 1st Qu.: 0.15509   1st Qu.: 0.14421   1st Qu.:  9.80
## Median : 0.19605   Median : 0.18945   Median : 38.34
## Mean   : 0.20378   Mean   : 0.20231   Mean   : 45.76
## 3rd Qu.: 0.24885   3rd Qu.: 0.24698   3rd Qu.: 70.23
## Max.   : 0.53831   Max.   : 0.54602   Max.   :390.60
## NA's   :22         NA's   :22         NA's   :13

## reanalysis_air_temp_k   reanalysis_avg_temp_k   reanalysis_dew_point_temp_k
## Min.   :294.6           Min.   :294.9           Min.   :289.6
## 1st Qu.:297.7           1st Qu.:298.3           1st Qu.:294.1
## Median :298.6           Median :299.3           Median :295.6
## Mean   :298.7           Mean   :299.2           Mean   :295.2
## 3rd Qu.:299.8           3rd Qu.:300.2           3rd Qu.:296.5
## Max.   :302.2           Max.   :302.9           Max.   :298.4
## NA's   :10              NA's   :10              NA's   :10

## reanalysis_max_air_temp_k   reanalysis_min_air_temp_k
## Min.   :297.8               Min.   :286.9
## 1st Qu.:301.0               1st Qu.:293.9
## Median :302.4               Median :296.2
## Mean   :303.4               Mean   :295.7
## 3rd Qu.:305.5               3rd Qu.:297.9
## Max.   :314.0               Max.   :299.9
## NA's   :10                  NA's   :10

## reanalysis_precip_amt_kg_per_m2   reanalysis_relative_humidity_percent
## Min.   :  0.00                    Min.   :57.79
## 1st Qu.: 13.05                    1st Qu.:77.18
## Median : 27.25                    Median :80.30
## Mean   : 40.15                    Mean   :82.16
## 3rd Qu.: 52.20                    3rd Qu.:86.36
## Max.   :570.50                    Max.   :98.61
## NA's   :10                        NA's   :10

## reanalysis_sat_precip_amt_mm   reanalysis_specific_humidity_g_per_kg
## Min.   :  0.00                 Min.   :11.72
## 1st Qu.:  9.80                 1st Qu.:15.56
## Median : 38.34                 Median :17.09
## Mean   : 45.76                 Mean   :16.75
## 3rd Qu.: 70.23                 3rd Qu.:17.98
## Max.   :390.60                 Max.   :20.46
## NA's   :13                     NA's   :10
```

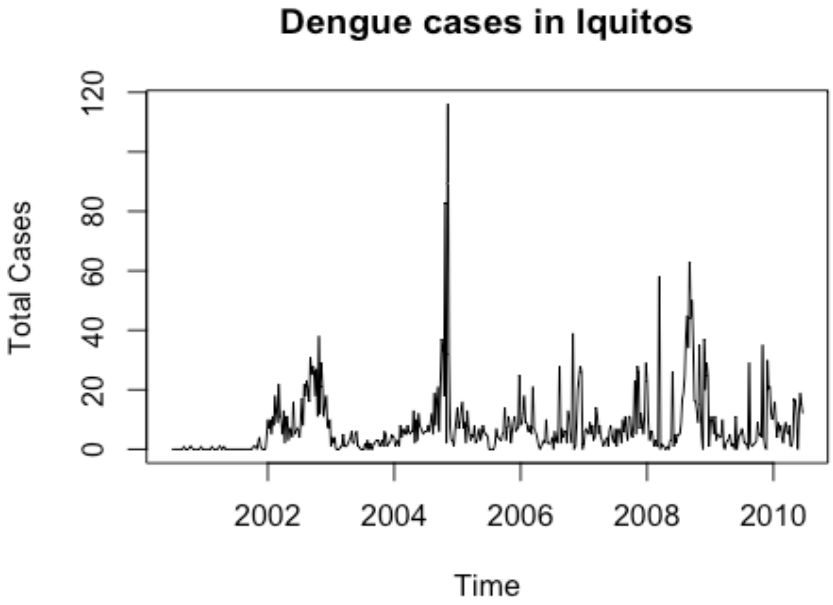
```
##  reanalysis_tdtr_k      station_avg_temp_c     station_diur_temp_rng_c
##  Min.   : 1.357         Min.   :21.40          Min.   : 4.529
##  1st Qu.: 2.329         1st Qu.:26.30          1st Qu.: 6.514
##  Median : 2.857         Median :27.41          Median : 7.300
##  Mean   : 4.904         Mean   :27.19          Mean   : 8.059
##  3rd Qu.: 7.625         3rd Qu.:28.16          3rd Qu.: 9.567
##  Max.   :16.029         Max.   :30.80          Max.   :15.800
##  NA's   :10             NA's   :43             NA's   :43

##  station_max_temp_c     station_min_temp_c     station_precip_mm
##  Min.   :26.70          Min.   :14.7           Min.   :  0.00
##  1st Qu.:31.10          1st Qu.:21.1           1st Qu.:  8.70
##  Median :32.80          Median :22.2           Median : 23.85
##  Mean   :32.45          Mean   :22.1           Mean   : 39.33
##  3rd Qu.:33.90          3rd Qu.:23.3           3rd Qu.: 53.90
##  Max.   :42.20          Max.   :25.6           Max.   :543.30
##  NA's   :20             NA's   :14             NA's   :22
```

Next, the dataset is split up into two datasets, based on the city. A dataset for San Juan is plotted below. It appears the dataset is both seasonal and cyclical, with an extreme peak near 1995.



Below is a plot of total number of dengue fever cases in Iquitos, Peru. The first thing noticed is that Iquitos' dengue outbreaks do not follow the same pattern as the ones presented for San Juan. This difference is possibly related to the two cities being in different hemispheres. In viewing the dengue cases in Iquitos from 2000 until 2011, we notice the same seasonal and cyclical effect. Also, there is an extreme spike in this dataset as well, although around 2005, 10 years after the spike present in the San Juan dataset.

```
plot(ts.iqtrain, main='Dengue cases in Iquitos', ylab='Total Cases')
```



Next, we look at the missing values present in the dataset. In the San Juan dataset, we notice that there some missing values present in the feature columns. The number of missing values is not consistent across the entire dataset, although some of the weather variables do have a consistent number of null values.

```
##Check to see number of null values in each column:
colSums(is.na(sjtrain))
```

```
##                    city                    49
##                       0                ndvi_se
##                    year                    19
##                       0                ndvi_sw
##              weekofyear                    19
##                       0     precipitation_amt_mm
##            total_cases                     9
##                       0  reanalysis_air_temp_k
##         week_start_date                     6
##                       0  reanalysis_avg_temp_k
##                ndvi_ne                     6
##                     191 reanalysis_dew_point_temp_k
##                ndvi_nw                     6
```

```
##          reanalysis_max_air_temp_k
##                                  6
##          reanalysis_min_air_temp_k
##                                  6
##      reanalysis_precip_amt_kg_per_m2
##                                  6
##  reanalysis_relative_humidity_percent
##                                  6
##        reanalysis_sat_precip_amt_mm
##                                  9
## reanalysis_specific_humidity_g_per_kg
##                                  6
##                reanalysis_tdtr_k
##                                  6
##             station_avg_temp_c
##                                  6
##          station_diur_temp_rng_c
##                                  6
##             station_max_temp_c
##                                  6
##             station_min_temp_c
##                                  6
##              station_precip_mm
##                                  6
```

Next we look at null values in the Iquitos dataset. Missing values are also present in the Iquitos dataset, although in different numbers. Another interesting find is that the missing values are not consistent between the San Juan dataset and the Iquitos dataset.

```
colSums(is.na(iqtrain))
##                          city
##                             0
##                          year
##                             0
##                    weekofyear
##                             0
##                   total_cases
##                             0
##                week_start_date
##                             0
##                        ndvi_ne
##                             3
##                        ndvi_nw
##                             3
##                        ndvi_se
##                             3
##                        ndvi_sw
##                             3
##             precipitation_amt_mm
##                             4
##             reanalysis_air_temp_k
##                             4
##             reanalysis_avg_temp_k
##                             4
##         reanalysis_dew_point_temp_k
##                             4
##          reanalysis_max_air_temp_k
##                             4
##          reanalysis_min_air_temp_k
##                             4
##      reanalysis_precip_amt_kg_per_m2
##                             4
##  reanalysis_relative_humidity_percent
##                             4
##        reanalysis_sat_precip_amt_mm
##                             4
## reanalysis_specific_humidity_g_per_kg
##                             4
##                reanalysis_tdtr_k
##                             4
##             station_avg_temp_c
##                            37
##          station_diur_temp_rng_c
##                            37
##             station_max_temp_c
##                            14
##             station_min_temp_c
##                             8
##              station_precip_mm
##                            16
```

Prior to imputing the dataset, we will create flag variables for missing values. The flag variables are created in a new column and for every null value, a '1' is placed in the new column, which has a prefix of 'm_'. If the value is present, a value of '0' is placed in the new column. The purpose of the flag variable is to allow the model to determine if the missing values were predictive in nature.

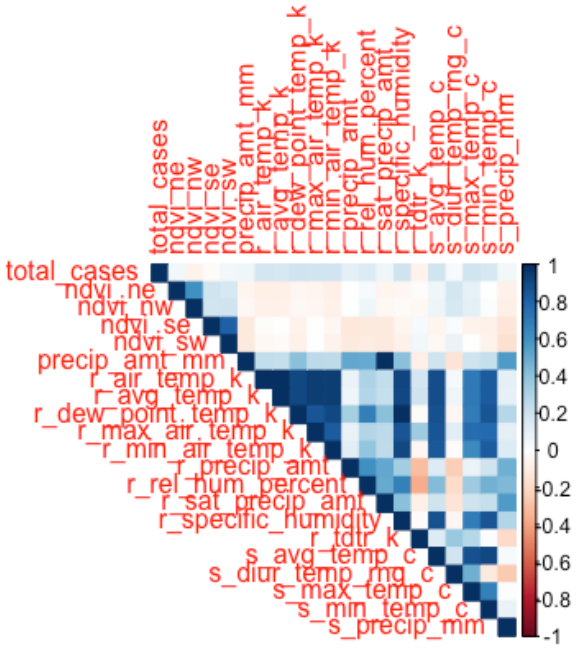
```
## create flags for null variables

sjtrain[ , paste0( "M_",names(sjtrain)[-1])] <-
      lapply(sjtrain[-1], function(x) as.numeric(is.na(x)) )

iqtrain[ , paste0( "M_",names(iqtrain)[-1])] <-
      lapply(iqtrain[-1], function(x) as.numeric(is.na(x)) )

## create flags for null variables

sjtest[ , paste0( "M_",names(sjtest)[-1])] <-
      lapply(sjtest[-1], function(x) as.numeric(is.na(x)) )

iqtest[ , paste0( "M_",names(iqtest)[-1])] <-
      lapply(iqtest[-1], function(x) as.numeric(is.na(x)) )
```

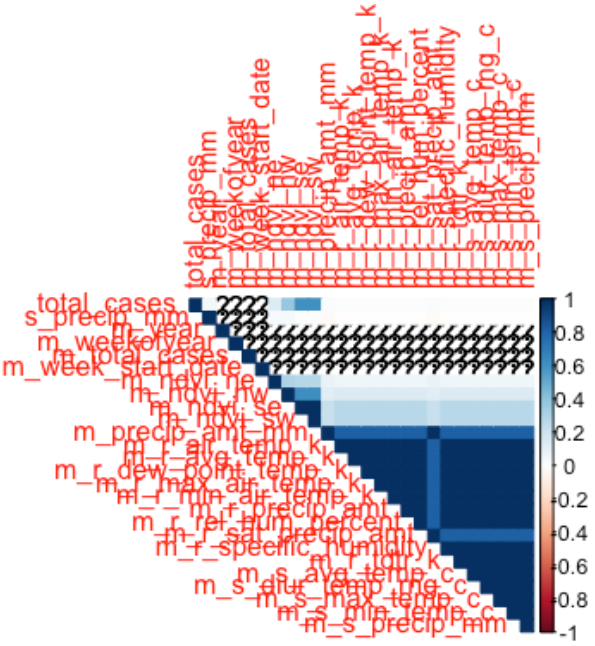
Now that we have a better idea what the dataset looks like and we've created flag variables for the missing values, we will address missing values as well as feature engineering prior to modeling. The missing values present in the San Juan and Iquitos dataset are imputed, not using a median or mean, but using the last value present in the dataset. Since the dengue dataset is a time series, by imputing with the last present value, we attempt to maintain the trend that was present in the dataset, at the time of the null value.

```
#replace null values with the previous value in each column
sjtrain<-na.locf(sjtrain)
iqtrain<-na.locf(iqtrain)

#test
#replace null values with the previous value in each column
sjtest<-na.locf(sjtest)
iqtest<-na.locf(iqtest)
```

Below we see the correlation plots for the San Juan variables in the dataset.

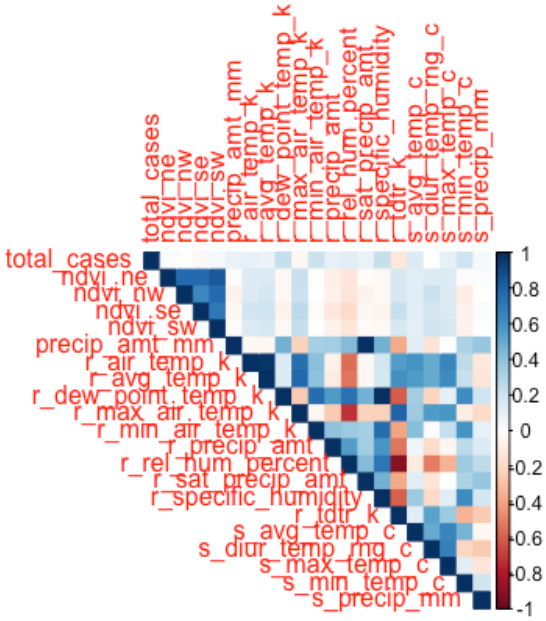
```
corrplot(sjcorr, method = "color", type = "upper")
corrplot(sjcorr1, method = "color", type = "upper")
```

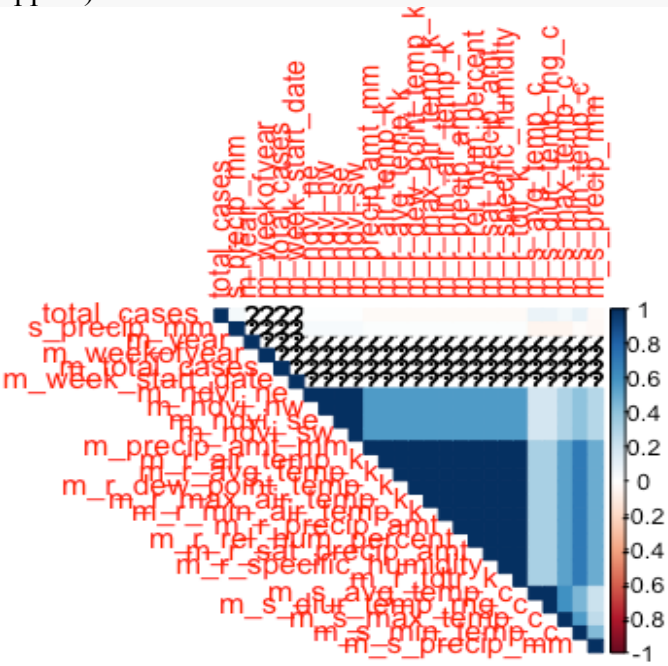




Below we see the same correlation plots, but for Iquitos.

```
corrplot(iqcorr, method = "color", type = "upper")
corrplot(iqcorr1, method = "color", type = "upper")
```



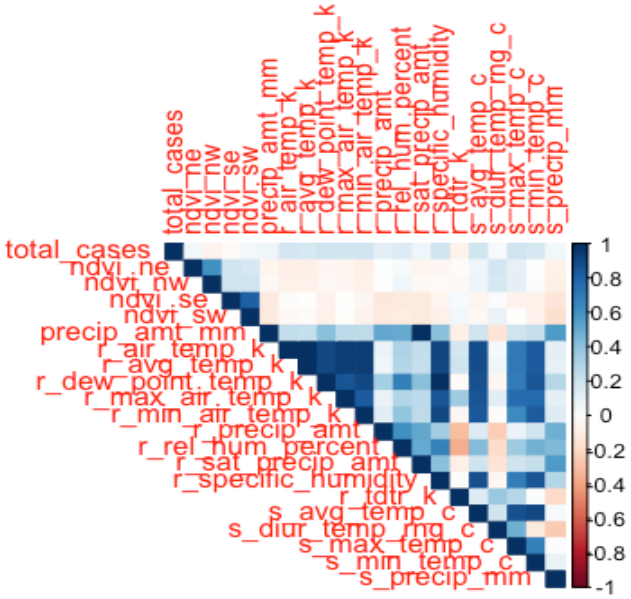


Now that we've created flag variables, dealt with null values in dataset and have a better idea of which variables are correlated with the target variable of 'total_cases', we will create lag variables for each of these existing variables. We choose to create variable lags of 1 week, 2 weeks, 3 weeks and 4 weeks. We are testing to see if the measured levels of the variables in previous weeks is more predictive than the current week on the total number of dengue cases each week. The lag period is indicated by the numerical suffix, '_1' meaning 1 week lag, '_2' meaning 2 weeks lag, etc.

Below we see the correlation plots for each lag variable created in relation to total_cases. San Juan is presented first, followed by Iquitos.

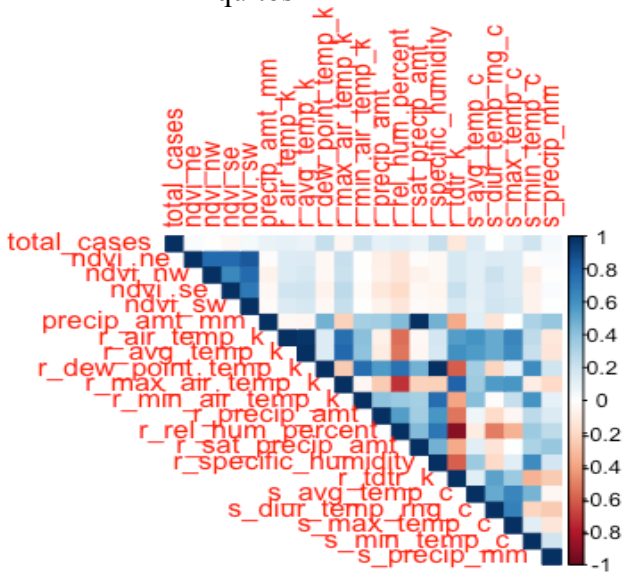
```
corrplot(sjcorr, method = "color", type = "upper")
```

San Juan



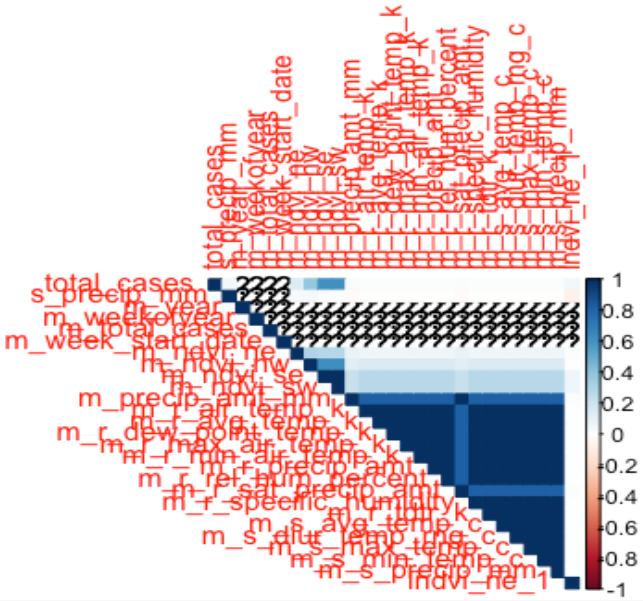

```
corrplot(iqcorr, method = "color", type = "upper")
```

Iquitos



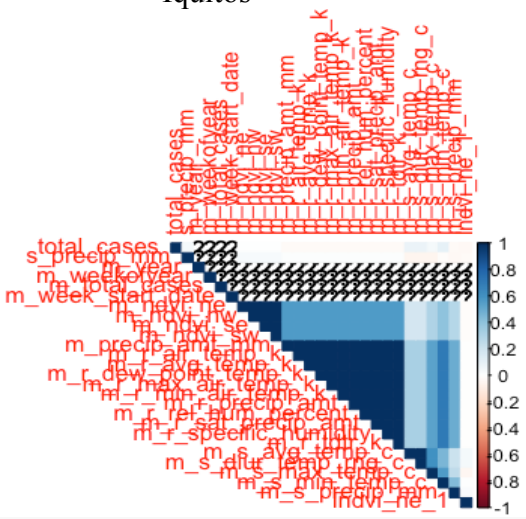
```
corrplot(sjcorr1, method = "color", type = "upper")
```

Sam Juan



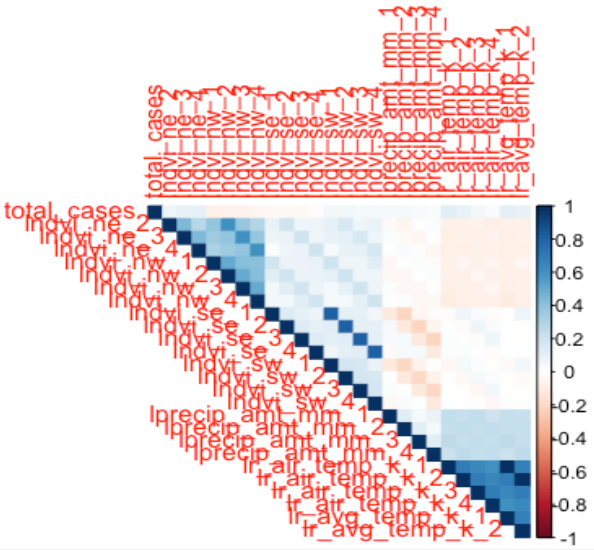

```
corrplot(iqcorr1, method = "color", type = "upper")
```

Iquitos



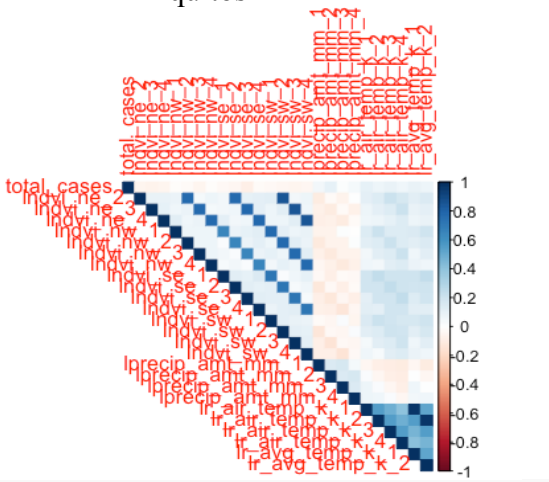

```
corrplot(sjcorr2, method = "color", type = "upper")
```

San Juan



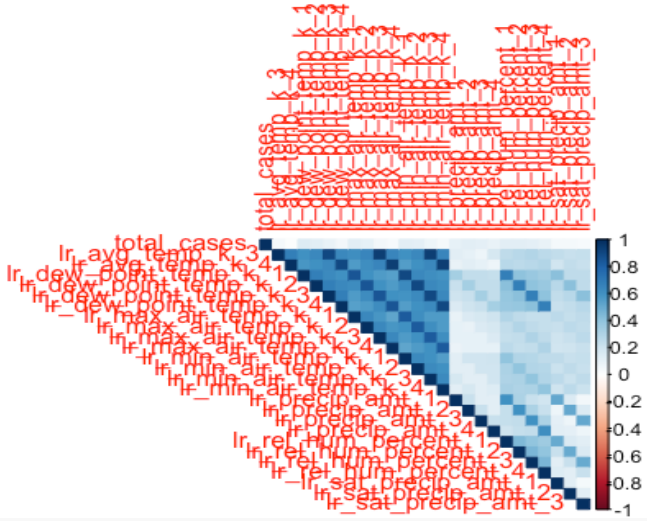

```
corrplot(iqcorr2, method = "color", type = "upper")
```

Iquitos



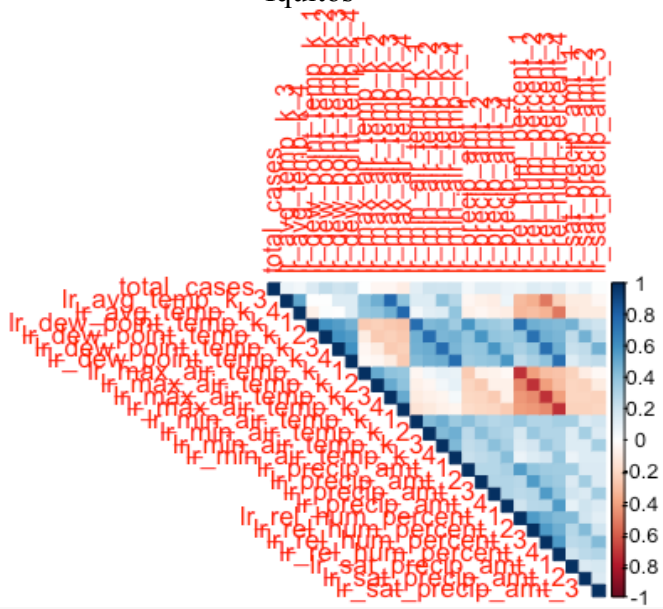

```
corrplot(sjcorr3, method = "color", type = "upper")
```

San Juan



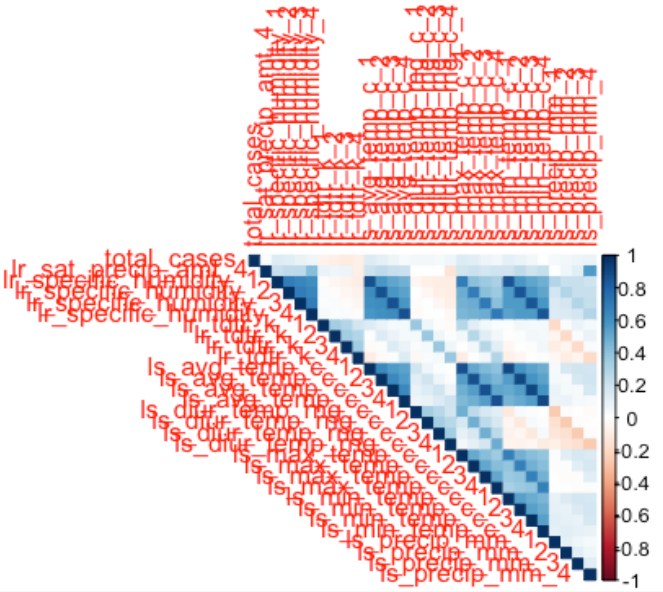

```
corrplot(iqcorr3, method = "color", type = "upper")
```

Iquitos



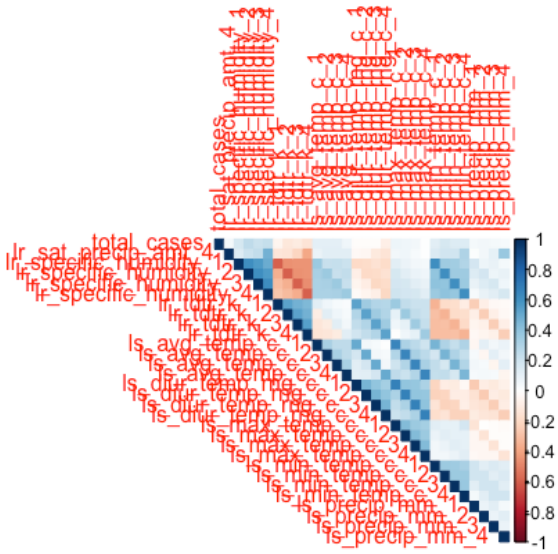

```
corrplot(sjcorr4, method = "color", type = "upper")
```

San Juan




```
corrplot(iqcorr4, method = "color", type = "upper")
```

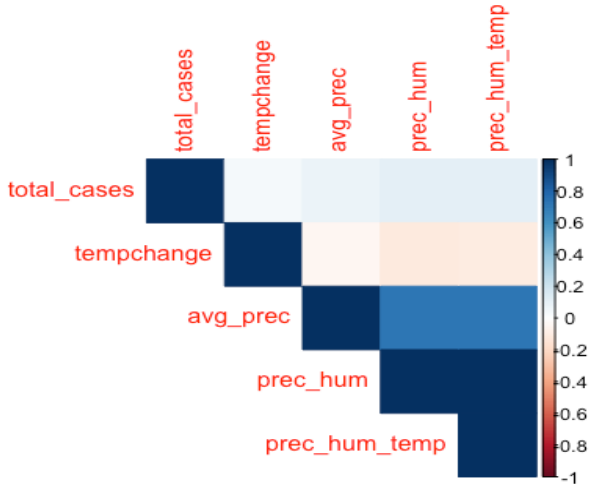
Iquitos



The last bit of feature engineering done is in creating some new variables: tempchange, avg_prec, prec_hum and prec_hum_temp. The variable tempchange is the max temperature in a week minus the minimum temperature in the same week. avg_prec is the sum of all 4 different precipitation measures, divided by 4, or the average precipitation, as measured by 4 different datasets. prec_hum is an interaction term between precipitation and humidity. Finally, prec_hum_temp is an interaction term between precipitation, humidity and temperature.
The correlation of these new variables is shown in the plots below. None of these variables appear to be particularly correlated with 'total_cases' in San Juan or Iquitos.

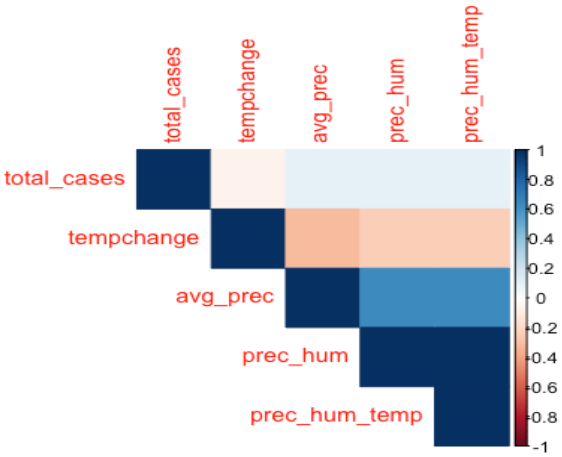
```
corrplot(sjcorr, method = "color", type = "upper")
```

San Juan




```
corrplot(iqcorr, method = "color", type = "upper")
```

Iquitos



Literature

While reviewing literature of analysis utilized to predict outbreaks of different diseases, the use of ARIMA models and ARIMAX models stood out. This research stretched from China, to Africa to Spain. In Spain, time-series analysis of Meningococcal disease was conducted in order to determine the positive and negative correlations with the outbreak of the disease (Dominguez et. al, 2007).

In China, research was conducted to identify the predictors of Hand Foot and Mouth Disease. Seasonal ARIMA models, as well as ARIMAX models incorporating weather data, were used to create highly accurate early warning systems (Song et. al, 2014). This same disease was the subject of research by Feng, Duan, Zhang and Zhang (2014), they also utilized a seasonal ARIMA model but did not feel their results were refined enough to be used in predicting future outbreaks of Hand, Foot and Mouth Disease. While these two papers had two different outcomes, while looking at the same disease, a main takeaway is the suitability of ARIMA and ARIMAX models to this type of problem. Another takeaway is the importance of providing variable importance or leading indicators, allowing public health officials to take action prior to an outbreak of the disease.

The above thought process is further supported by Medina, Findley, Guindo and Doumbia, in their research paper detailing their attempts to forecast diseases in Mali. In this paper, the authors state “public health programs therein could benefit from parsimonious general-purpose forecasting methods to enhance infectious disease intervention” (2007). In attempting to forecast the spread of infectious diseases, Medina and associates utilized a multiplicative Holt-Winters method.

Sato (2013) provides examples of other successful disease management forecasting using ARIMA models. Some of these include: “prediction of the number of beds occupied during the epidemic of severe acute respiratory syndrome (SARS) at a hospital in Singapore” and “predict and study antimicrobial resistance” (Sato, 2013).

Type of Models

In determining what models to use to predict total cases of dengue fever by week in San Juan and Iquitos, I felt the need to analyze using time series specific models, such as ARIMA and NNAR. These models were selected for their ability to model seasonal and cyclical trends, which is present in both the San Juan data and the Iquitos data.

Another model I chose to use in addressing this problem was a Random Forest. Since there are a large number of variables with an impact on the total number of dengue cases, the possibility existed for the random forest to identify relationships and decision branches that other models may not be as sensitive to.

Finally, I chose to use an Ordinary Least Squared models, as it provides a good benchmark model and is a good place to start. If the OLS model adequately models the relationship, there is no need to attempt other models, or complicate the analysis.

Formulation

First, I implemented the OLS Model in Python. In the OLS Model, I had to be very careful about collinearity as many of the variables in the dataset have a higher correlation coefficient with other variables than to the target variable of 'total_cases'. Due to the concern about collinearity, only a few variables were maintained in each model. Another thing of note is the difference between the models for San Juan and Iquitos. These differences are due to the impact of different variables on the total dengue cases each week in each city. The code for the OLS regression can be found below.

```
#San Juan
X_sj=sjtrain[['year',
 'weekofyear',
            'm_ndvi_se'
           ]]
X_sj['intercept']=1
y_sj=sjtrain[['total_cases']]

lm1 = sm.OLS(y_sj,X_sj).fit()
lm1.summary()

#Iquitos
X_iq=iqtrain[['year',
           's_min_temp_c',
            'lr_specific_humidity_2']]
X_iq['intercept']=1
y_iq=iqtrain[['total_cases']]

lm2 = sm.OLS(y_iq,X_iq).fit()
lm2.summary()
```

Next, I implemented an ARIMA model. This model did not include any outside variables and looked only at the total cases by city over time. ARIMA models are well suited for complex time series problems, but the lack of any external variables in this model leads me to believe that the ARIMA model won't perform as well as an ARIMAX model or other models which include the other variables present in the dataset.

In the below summary of the model, we see the ARIMA model for San Juan. It has a weekly seasonal effect, as indicated by the annotation [52] under the header "Forecast method".

```
sjarima<-forecast(auto.arima(ts.sjtrain),h=260)
summary(sjarima)
##
## Forecast method: ARIMA(4,1,5)(0,0,2)[52] with drift
##
## Model Information:
## Series: ts.sjtrain
## ARIMA(4,1,5)(0,0,2)[52] with drift
##
## Coefficients:
##          ar1      ar2      ar3      ar4     ma1      ma2     ma3     ma4
##      -0.8574  -0.1761  -0.5594  -0.7157  0.4196  -0.1804  0.5120  0.6525
## s.e.  0.0773   0.1095   0.1046   0.0605  0.0850   0.0806  0.0586  0.0429
##          ma5     sma1    sma2   drift
##      -0.2080  -0.0716  0.0785  0.0024
## s.e.  0.0487   0.0356  0.0333  0.5362
##
## sigma^2 estimated as 615.2:  log likelihood=-4323.8
## AIC=8673.61   AICc=8674   BIC=8736.54
##
## Error measures:
##                       ME             RMSE      MAE        MPE    MAPE     MASE       ACF1
## Training set 0.001792978    24.63093   12.03565    -Inf      Inf        0.327118    0.001478486
```

In the ARIMA model for Iquitos, we notice that there is not a seasonal component, like the one we saw in the San Juan model. This is very unexpected. This major difference between the two datasets is very confusing.

```
iqarima<-forecast(auto.arima(ts.iqtrain),h=156)
summary(iqarima)
##
## Forecast method: ARIMA(0,1,3)
##
## Model Information:
## Series: ts.iqtrain
## ARIMA(0,1,3)
##
```

```
## Coefficients:
##          ma1     ma2      ma3
##       -0.7712  0.2107  -0.1688
## s.e.   0.0438  0.0560   0.0467
##
## sigma^2 estimated as 79.97:  log likelihood=-1872.33
## AIC=3752.67   AICc=3752.75   BIC=3769.68
##
## Error measures:
##                     ME             RMSE      MAE      MPE     MAPE      MASE
ACF1
## Training set 0.08380256   8.907887   4.80042   -Inf      Inf          0.5089707   0.004615791
```

Next, I implemented an ARIMAX model. This model uses the capability of the ARIMA model to model time series data and allows for the inclusion of external variables. For the San Juan ARIMAX model, we use the flag variable 'm_ndvi_se', which captures the impact of a missing value in the 'ndvi_se' column. This variable was chosen because of its high correlation with 'total_cases' in San Juan of 0.59. Below are the summary results.

The ARIMA portion of the model has changed now that an exogenous variable has been added. Despite this our MAE, or mean absolute error, has barely gone down. This model may be better than the ARIMA model for San Juan, but not in a noticeable way. In the Performance/Accuracy section of this report, we will explore this model's relation to other models in more detail.

```
sj_arimax<-forecast(auto.arima(ts.sjtrain, xreg= sjtrain$m_ndvi_se),xreg= sjtrain$m_ndvi_se,
h=260)
summary(sj_arimax)
#
#Forecast method: Regression with ARIMA(0,1,1) errors
#
#Model Information:
#Series: ts.sjtrain
#Regression with ARIMA(0,1,1) errors
#
#Coefficients:
#         ma1     xreg
#    -0.4102  85.4685
#s.e.   0.0287   7.3444
#
#sigma^2 estimated as 569.5:  log likelihood=-4292.01
#AIC=8590.01   AICc=8590.04   BIC=8604.53
#
#Error measures:
#                    ME                RMSE      MAE          MPE     MAPE      MASE
```

```
#Training set 0.003177345     23.82675   12.03461    -Inf       Inf          0.3270897
#                   ACF1
#Training set -0.01265279
```

For the Iquitos ARIMAX model, the variable with the highest correlation coefficient was chosen. This variable was lr_specific_humidity_2. This variable is the 2-week lag variable of r_specific_humidity. The results of the ARIMAX model are presented below. In looking at the below summary, we notice that the ARIMA model has changed. Also, the MAE is higher in this model than in the straight ARIMA model. This was unexpected, as the general thought was that adding outside factors to the dataset would provide a more accurate forecast.

```
iq_arimax<-forecast(auto.arima(ts.iqtrain, xreg=iqtrain$lr_specific_humidity_2),
xreg=iqtrain$lr_specific_humidity_2, h=156)
summary(iq_arimax)
#Forecast method: Regression with ARIMA(2,0,1) errors
#
#Model Information:
#Series: ts.iqtrain
#Regression with ARIMA(2,0,1) errors
#
#Coefficients:
#         ar1     ar2      ma1     xreg
#      0.5289  0.2828  -0.3333  0.4564
#s.e.  0.0987  0.0637   0.1022  0.0764
#
#sigma^2 estimated as 76.7:  log likelihood=-1864.5
#AIC=3739.01   AICc=3739.12   BIC=3760.27
#
#Error measures:
#                    ME              RMSE        MAE          MPE     MAPE          MASE
#Training set -0.04714346     8.723872     4.939147     -Inf       Inf           0.5236794
#                   ACF1
#Training set 0.002280775
```

After running the ARIMAX models, I ran a NNAR model, or a Neural Net Auto Regressive model. We provide the neural net with all of the data and allow it to build the best network. Below is the code from the models.

While neural nets are known to be highly accurate, they lack interpretability. If an accurate prediction is the only thing that matters, this is fine. In most cases, interpretability is of paramount importance. In the case of the current problem, predicting the number of cases, while the goal of the analysis, does not help scientists, doctors and researchers in the area. Without an understanding of why the number of cases of dengue is increasing, these personnel aren't able to educate the population or take necessary precautions to prevent an increase in cases. Despite the NNAR likely not being the most appropriate model for this

problem in a real-world setting, in the current state, we will use it for its increased accuracy. The accuracy metrics of this model will be shown in the Performance/Accuracy section of this report.

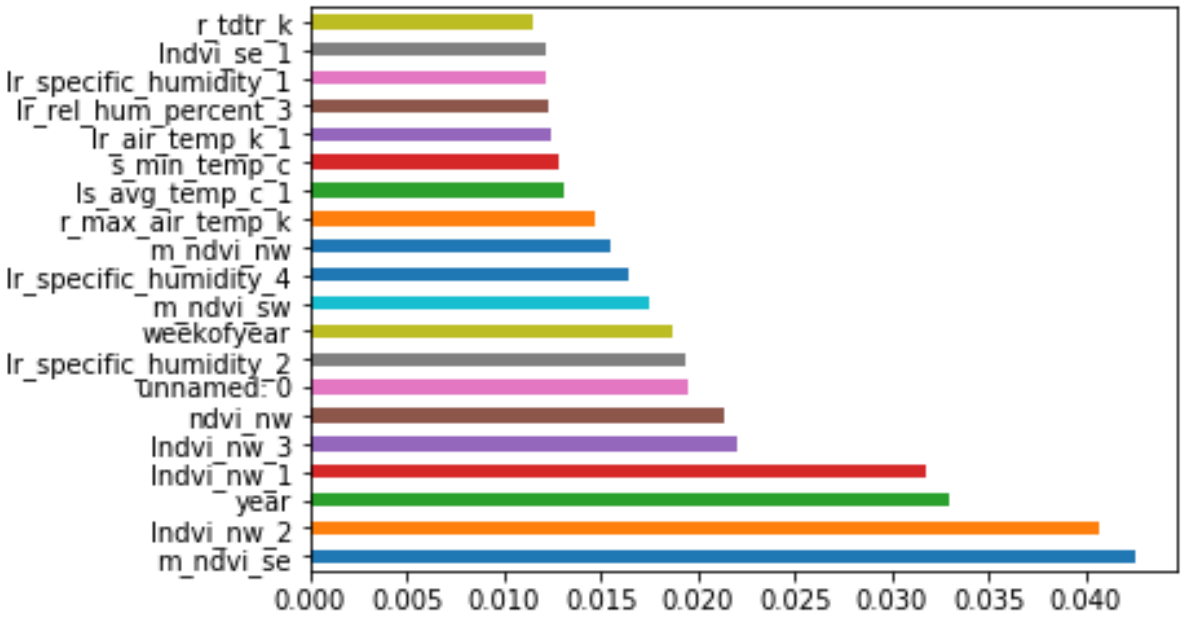
```
nntsj<-nnetar(ts.sjtrain)
nntiq<-nnetar(ts.iqtrain)
```

Finally, I implemented a Random Forest and tuned the hyper parameters. A random forest is, simply put, a forest of decision trees. By determining the different decision points, a flexible and accurate model can be built. While random forest results are not as interpretable as an OLS model, there is some interpretability that can be taken from the model in the form of feature importance. This is an output that shows which variables have the largest impact on the target variable, in this case 'total_cases'. This model is an appropriate choice, both in an accuracy standpoint and an interpretability standpoint, because the scientists, doctors and researchers working on this problem would be able to determine which factors have the largest impact, and thus can account for these factors in their attempts to prevent an increase in the number of dengue fever cases in San Juan and Iquitos. The code used to identify the optimum parameters and build the random forest in Python can be found below:

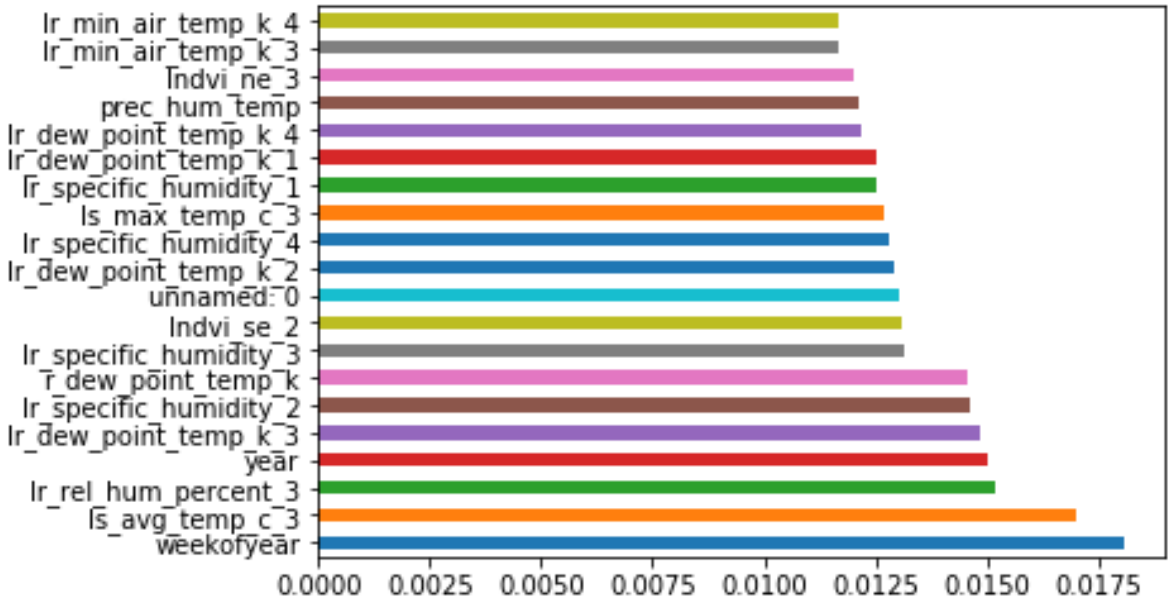
```
from sklearn.model_selection import GridSearchCV

# Create the parameter grid based on the results of random search
param_grid = {
    'bootstrap': [True],
    'max_depth': [80, 90, 100, 110],
    'max_features': [2, 3],
    'min_samples_leaf': [3, 4, 5],
    'min_samples_split': [8, 10, 12],
    'n_estimators': [100, 200, 300, 1000]
}
# Create a based model
rf = RandomForestRegressor()
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid,
                           cv = 3, n_jobs = -1, verbose = 2)

# Fit the grid search to the data
grid_search.fit(sjfeatures, sjlabels)
grid_search.best_params_

# Instantiate model with 1000 decision trees
rfsj = RandomForestRegressor(n_estimators=100,  max_depth=100, min_samples_split=8,
min_samples_leaf=5,  max_features=2,   bootstrap=True, random_state=42)
# Train the model on training data
rfsj.fit(sjfeatures, sjlabels);
```

```
# Use the forest's predict method on the test data
sjpredictions = rfsj.predict(sjfeatures_test)


##Iquitos
# Fit the grid search to the data
grid_search.fit(iqfeatures, iqlabels)
grid_search.best_params_

rfiq = RandomForestRegressor(n_estimators=200,  max_depth=90, min_samples_split=8, min_s
amples_leaf=4,  max_features=2,   bootstrap=True, random_state=42)

# Train the model on training data
rfiq.fit(iqfeatures, iqlabels);
# Use the forest's predict method on the test data
iqpredictions = rfiq.predict(iqfeatures_test)

feat_importances = pd.Series(rfsj.feature_importances_, index=sjfeature_list)
feat_importances = feat_importances.nlargest(20)
feat_importances.plot(kind='barh')

feat_importances = pd.Series(rfiq.feature_importances_, index=iqfeature_list)
feat_importances = feat_importances.nlargest(20)
feat_importances.plot(kind='barh')
```

Below is the feature importance for San Juan. The feature with the largest impact on 'total_cases' is 'm_ndvi_se', which is an indicator variable for missing variables in the 'm_ndvi_se' variable. After 'm_ndvi_se', the variable 'lndvi_nw_2' has the largest impact. This is a 2-week lag variable for 'ndvi_nw'. The rest of the important features can be seen in the below image.



The below image provides the feature importance for Iquitos. A main takeaway in comparing this feature importance to the one for San Juan is the almost complete difference in features important to predicting total cases of dengue fever. The week of the year is the most important feature for Iquitos, followed by a 3-week lag variable of the average temperature.



Performance/Accuracy

Below is the summary output from the San Juan OLS Model. First thing of note is the R-squared. At 0.412, the model only accounts for approximately 41% of the variation present in the dataset. This is not ideal.

OLS Regression Results

Dep. Variable:	total_cases	**R-squared:**	0.412
Model:	OLS	**Adj. R-squared:**	0.410
Method:	Least Squares	**F-statistic:**	217.9
Date:	Thu, 07 Jun 2018	**Prob (F-statistic):**	4.50e-107
Time:	11:16:59	**Log-Likelihood:**	-4766.1
No. Observations:	936	**AIC:**	9540.
Df Residuals:	932	**BIC:**	9560.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
year	-1.4215	0.249	-5.704	0.000	-1.911	-0.932
weekofyear	0.6269	0.087	7.181	0.000	0.456	0.798
m_ndvi_se	200.7382	9.316	21.547	0.000	182.455	219.021
intercept	2854.8953	498.316	5.729	0.000	1876.944	3832.846

Omnibus:	414.812	**Durbin-Watson:**	0.604
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	8242.126
Skew:	1.528	**Prob(JB):**	0.00
Kurtosis:	17.213	**Cond. No.**	7.72e+05

Next, we can see the OLS model summary for Iquitos. If we were unimpressed with 0.412, the 0.085 R-squared for the Iquitos OLS model is laughable. These OLS models are not good models for predicting the total cases of dengue fever.

OLS Regression Results

Dep. Variable:	total_cases	**R-squared:**	0.085
Model:	OLS	**Adj. R-squared:**	0.079
Method:	Least Squares	**F-statistic:**	15.93
Date:	Thu, 07 Jun 2018	**Prob (F-statistic):**	6.39e-10
Time:	11:17:16	**Log-Likelihood:**	-1950.0
No. Observations:	520	**AIC:**	3908.
Df Residuals:	516	**BIC:**	3925.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
year	0.4078	0.164	2.482	0.013	0.085	0.731
s_min_temp_c	1.0471	0.394	2.661	0.008	0.274	1.820
ir_specific_humidity_2	1.2026	0.357	3.368	0.001	0.501	1.904
intercept	-852.8165	327.700	-2.602	0.010	-1496.607	-209.026

Omnibus:	485.193	**Durbin-Watson:**	1.217
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	18360.355
Skew:	4.015	**Prob(JB):**	0.00
Kurtosis:	30.980	**Cond. No.**	1.45e+06

Below we see the accuracy metrics for the different models executed in R. In order: ARIMA (San Juan), ARIMA (Iquitos), ARIMAX(San Juan), ARIMAX (Iquitos), NNAR (San Juan) and NNAR (Iquitos). Highlighted in yellow is the mean absolute error (MAE). This metric is the evaluation metric on DrivenData and thus the training MAE is highlighted for comparison.

In looking at the MAE, the Iquitos MAE is consistently lower than the San Juan dataset. In thinking about our ARIMA models, and the lack of seasonal trend in the Iquitos dataset, the Iquitos dataset may be simpler in terms of predicting. Unfortunately, we aren't just predicting San Juan or Iquitos, the MAE of both datasets needs to be minimized. Unsurprisingly, the NNAR models have the lowest MAE for both San Juan and Iquitos.

```
accuracy(sjarima)
##                    ME     RMSE      MAE  MPE MAPE     MASE        ACF1
## Training set 0.001792978 24.63093 12.03565 -Inf  Inf 0.327118 0.001478486
accuracy(iqarima)
##                    ME     RMSE     MAE  MPE MAPE     MASE      ACF1
## Training set 0.08380256 8.907887 4.80042 -Inf  Inf 0.5089707 0.004615791
accuracy(sj_arimax)
##                    ME     RMSE      MAE  MPE MAPE     MASE       ACF1
## Training set 0.003177345 23.82675 12.03461 -Inf  Inf 0.3270897 -0.01265279
accuracy(iq_arimax)
```

```
##                      ME             RMSE      MAE       MPE      MAPE      MASE         ACF1
## Training set -0.04714346 8.723872 4.939147 -Inf Inf 0.5236794 0.002280775
accuracy(nntsj)
##                      ME             RMSE      MAE       MPE      MAPE      MASE         ACF1
## Training set -0.06567217 10.28473 6.128826 -Inf Inf 0.1665759 -0.09451233
accuracy(nntiq)
##                      ME             RMSE      MAE       MPE      MAPE      MASE         ACF1
## Training set -0.01802247 3.303664 2.25782 NaN Inf 0.2393883 -0.004654597
```

While the training accuracy is nice to see, it can suffer from overfitting and can misrepresent the accuracy of the model. Thus, using a test set to collect accuracy metrics is necessary. Driven Data held out a test set and thus by submitting the predictions for this test set, we are able to see which models the best predictors of total cases of dengue fever in San Juan and Iquitos are. The below table provides the different models used and the mean absolute error, as reported by Driven Data.

The random forest with the tuned hyper parameters is the most accurate model when predicting the total cases of dengue fever in San Juan and Iquitos.

Model	Driven Data MAE
OLS	34.5505
ARIMA	34.1394
ARIMAX	33.5817
NNAR	26.8582
RF	26.2019
Submitted 0s	37.6010

Limitations

Some of the limitations present in this analysis are related to the lack of subject matter expertise possessed by the analyst, namely myself. In looking at the dataset provided, I am able to take a guess at which variables are most important, and also use correlation matrices, and the like, to get a numerical estimate of what variables are the most important. Despite this, a lack of understanding prevents me from engineering good features that are not currently contained in the dataset.

Another limitation present is the use of only the data provided by Driven Data. By incorporating outside datasets, such as holiday schedules in San Juan and Iquitos, or indicators for previous preventative measures implemented, the model would likely become more accurate.

Finally, the difference in accuracy between San Juan and Iquitos is a limitation in the use of the model. By separating the models, Iquitos' total cases of dengue fever can be predicted with relative accuracy, whereas the mean absolute error for San Juan is a bit too high for personal comfort. This difference in accuracy is a limitation of the model and its implementation.

Future Work

In the future, I would incorporate outside data, such as holiday schedules, and indicators for different preventative measures that were implemented at different times. I would also attempt to learn more about dengue fever, as well as San Juan and Iquitos so that I could better engineer features to be used in modeling. Also, I'd take the knowledge of the dataset learned during this project to attempt more complex and accurate models. I'd be interested in seeing how XGBoost compared to the current models. I'd also take the opportunity to utilize a Long Short Term Neural Net (LSTM) and compare its accuracy to the NNAR model used during this analysis.

Learning

In working through this project, a major learning was the tradeoff between accuracy and interpretability. By thinking about who the end customer would be, and how they would interact with the results and predictions, better models were chosen. An example of this is the choice to model using a random forest, ensuring that if this were the real world, the results would be valuable and usable to those in the field.

Another learning was related to the lack of increased accuracy provided by the ARIMAX model over the ARIMA model. I am not sure if this lack of accuracy is unique to this issue, or if there is a threshold of correlation necessary for the exogenous variable to impact the predictions in a meaningful way.

Third, I was able to successfully use a random forest which I was unable to do during the mid-term. On top of this, I was able to tune the parameters, using a random grid search, resulting in an increase in accuracy on the results of the random forest. This use of machine learning to solve the problem was an achievement for me.

Finally, my learnings throughout this course allowed me to look at the problem and immediately identify it as a time series problem and to go to my "time series toolkit" to begin analyzing the data and building models. Without the knowledge and learnings gained in this class, the problem would have seemed daunting. Despite not choosing a time series model taught in the class for the final model, the application of time series techniques allowed me to better understand how the different cities were impacted and to begin addressing the problem at hand.

References

Domínguez, Muñoz, Cardeñosa, Martínez, & Caylà. (2007). Time-Series Analysis of Meningococcal Disease in Catalonia. Annals of Epidemiology, 17(9), 654-662.

Huifen Feng, Guangcai Duan, Rongguang Zhang, & Weidong Zhang. (n.d.). Time series analysis of hand-foot-mouth disease hospitalization in Zhengzhou: Establishment of forecasting models using climate variables as predictors. PLoS ONE, 9(1), E87916.

Medina DC, Findley SE, Guindo B, Doumbia S (2007) Forecasting Non-Stationary Diarrhea, Acute Respiratory Infection, and Malaria Time-Series in Niono, Mali. PLOS ONE 2(11): e1181. https://doi.org/10.1371/journal.pone.0001181

Sato, R. (2013). Disease management with ARIMA model in time series. Einstein, 11(1), 128-131.

Yuanbin Song, Fan Wang, Bin Wang, Shaohua Tao, Huiping Zhang, Sai Liu, . . . Qiyi Zeng. (2015). Time series analyses of hand, foot and mouth disease integrating weather variables. PLoS ONE, 10(3), E0117296.