Predictive Analytics & Forecasting - Final

Eric Kenney

2023-08-12

Abstract

Dengue Fever is a mosquito-borne tropical disease. In most cases it presents with flu-like symptoms, but in severe cases it can lead to death. Historically, dengue fever is prevalent in southeast Asia, but due to climate change, is being found more commonly in Africa and Latin America. Forecasting cases is useful to public health officials as they make decisions on resourcing and future needs. I create three models to forecast dengue fever on the training set, splitting it 80/20 for a hold out validation set. I use an ARIMA (1,1,3) model as a benchmark. Additionally, I create an ARIMA w/ Errors model along with a Neural Network. Both of these last two models using differenced and lagged values of three predictors (average air temperature, dew point temperature, and precipitation). On the validation data the ARIMA w/ Errors model performed the best, but the Neural Network performed the best on the test data with an MAE of 28.2 (the best model in the competition has an MAE of 10.1.

Problem Statement

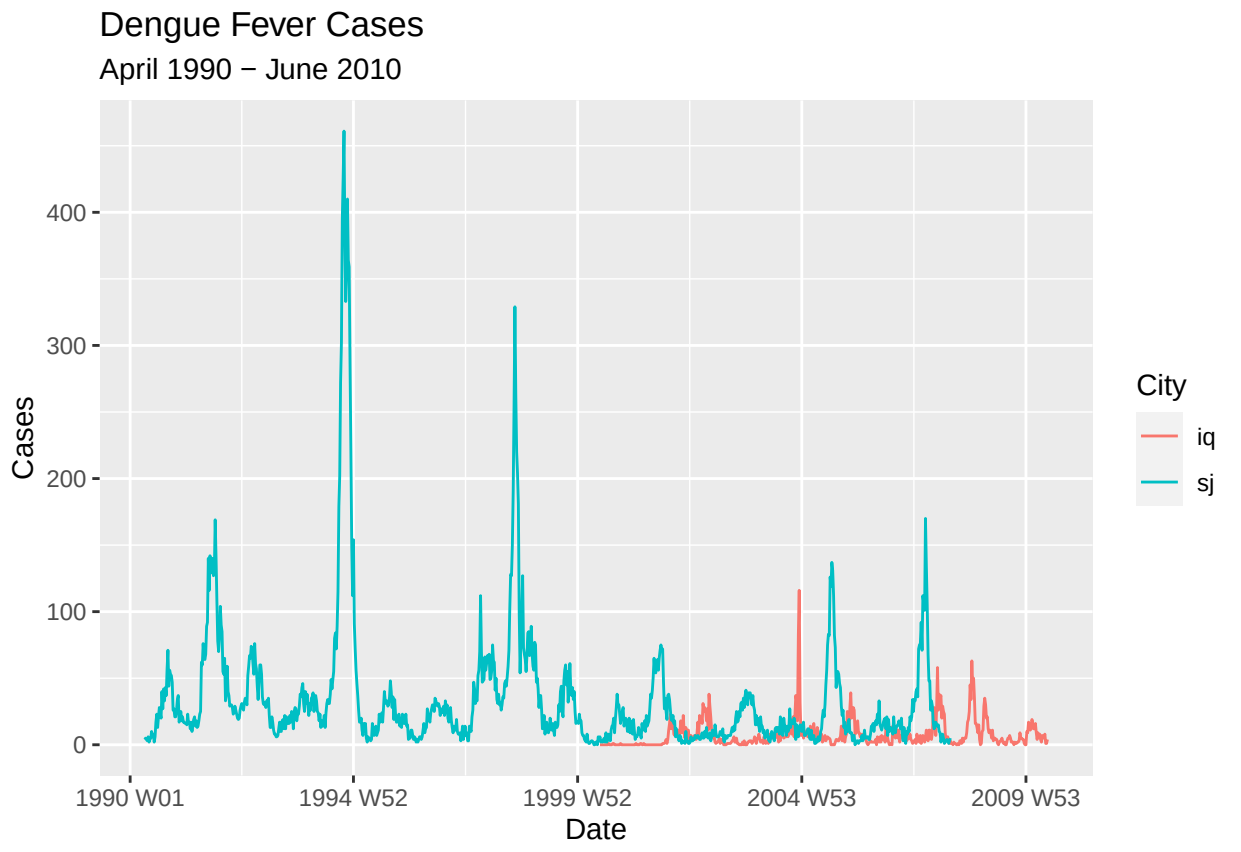
Using data from the DengAI: Predicting Disease Spread project from DrivenData I model and predict weekly cases of dengue fever in the cities of San Juan, Puerto Rico and Iquitos, Peru. Dengue fever is a tropical disease spread by mosquitos. In most cases, it resembles the flu with symptoms that include fever, rash, and muscle/joint pain which takes on average 2-7 days to recover. In more severe cases, dengue fever develops into dengue hemorrhagic fever or dengue shock syndrome which may lead to death. Historically, dengue fever was prevalent in Southeast Asia and the Pacific Islands. In recent years, more cases are being seen in Africa and Latin America. With climate change ever present, there is concern that shifts will continue to occur, leading to public health implications. While not a particularly deadly disease (0.8%-2.5% risk of death in severe cases), it does have the potential to utilize resources on an already strained health care system. Accurate modeling and forecasting can help public health officials prepare for future cases. The goal is to create a model that can apply across multiple cities and is not limited to just one.

Data Set

There are three data sets provided by DrivenData: *dengue_features_train*, *dengue_features_test*, *dengue_labels_train*. *dengue_features_train* and *dengue_features_test* contain 20 environmental variables for the San Juan and Iquitos for the weeks studied in the training set. *dengue_labels_train* contains the weekly total cases for each city. I split the training set into each city and decompose separately.

```
# Read in data
dengue_features_train = read.csv(file.path(path, "dengue_features_train.csv"))
dengue_features_test = read.csv(file.path(path, "dengue_features_test.csv"))
dengue_labels_train = read.csv(file.path(path, "dengue_labels_train.csv"))
dengue_labels_test = read.csv(file.path(path, "submission_format.csv"))
```

```
# Create Tsibble and combine training data
dengue_train = left_join(dengue_features_train, dengue_labels_train,
                         by = c("city", "year", "weekofyear"))

dengue_train = dengue_train %>%
  mutate(week = yearweek(base::as.Date(week_start_date))) %>%
  as_tsibble(index = week, key = city)

# Plot total cases
dengue_train %>%
  autoplot(total_cases) +
  labs(x = "Date", y = "Cases",
       title = "Dengue Fever Cases",
       subtitle = "April 1990 - June 2010") +
  guides(color = guide_legend(title = "City"))
```

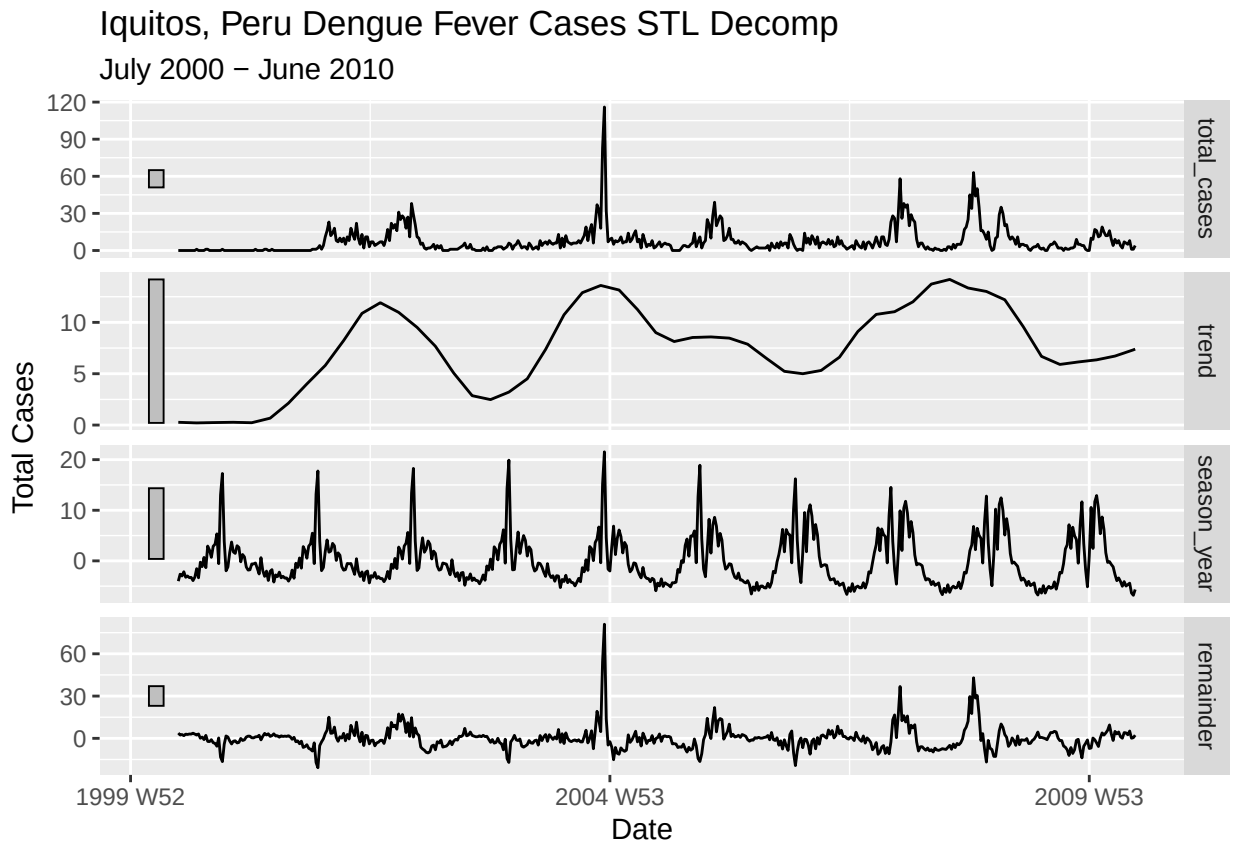



```
# Create training sets based on city
dengue_iq_train = dengue_train %>%
  filter(city == "iq")

dengue_sj_train = dengue_train %>%
  filter(city == "sj")

# Set Lambda for Box-Cox Transforms
lambda = 0.35
```

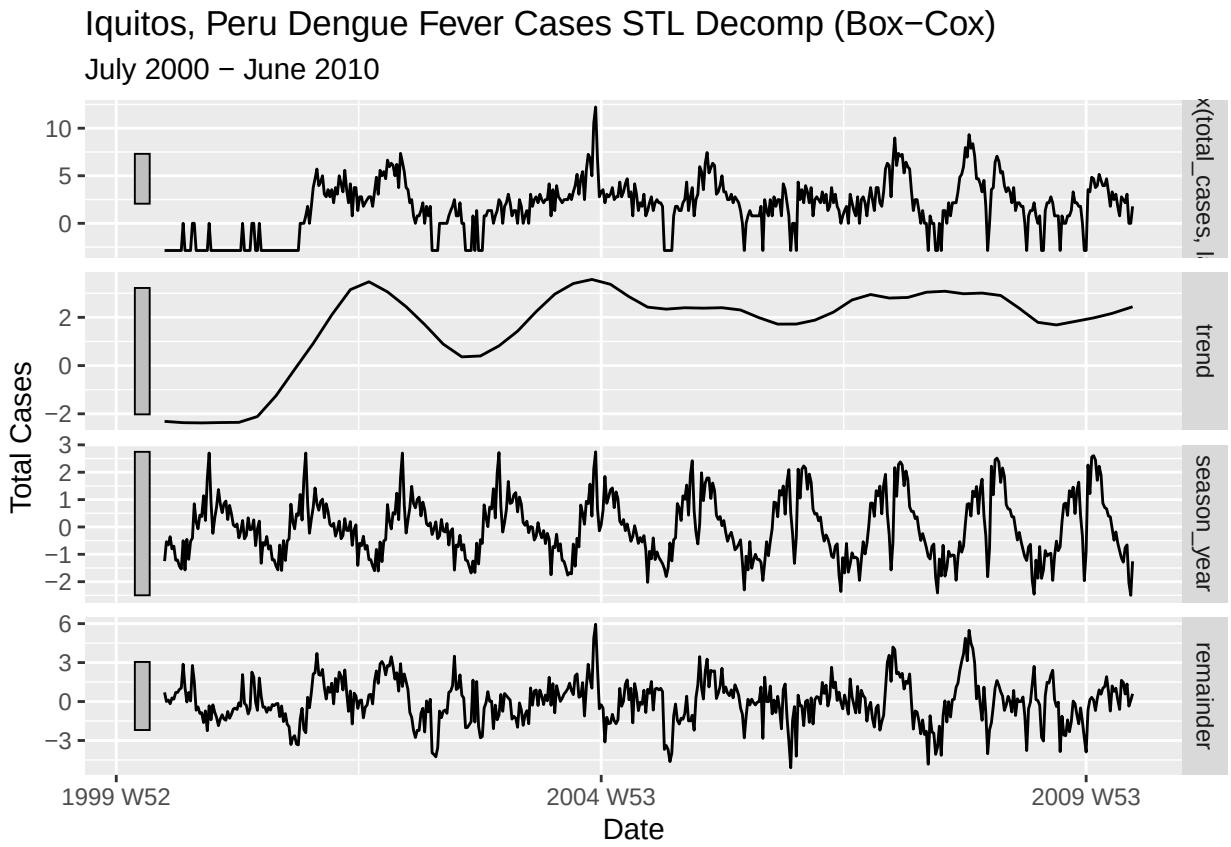
```
# Fill gaps and copy preceding value into it (2 missing values)
dengue_iq_train = dengue_iq_train %>%
  fill_gaps() %>%
  mutate_all( ~ na.locf(.x, na.rm = FALSE))

# Fill gaps and copy preceding value into it (three missing values)
dengue_sj_train = dengue_sj_train %>%
  fill_gaps() %>%
  mutate_all( ~ na.locf(.x, na.rm = FALSE))

# Grab lengths of training data for splits
train_iq_obs = dim(dengue_iq_train)[1]
train_sj_obs = dim(dengue_sj_train)[1]

# Grab 3 observations for differencing the test set later
extra_iq_train = tail(dengue_iq_train, 3)
extra_sj_train = tail(dengue_sj_train, 3)
```

There is ~16 years of data from San Juan and 10 years of data from Iquitos. There are a notably higher number of cases in Puerto Rico vs. Peru. Next, I look at STL decompositions of the data. If I apply any transformation to data in one city I apply the same in the other city as I am making a model to be applied regardless of location.

```
# Decompose Iquitos
## Untransformed
dengue_iq_train %>%
  model(STL(total_cases)) %>%
  components() %>%
  autoplot() +
  labs(x = "Date", y = "Total Cases",
       title = "Iquitos, Peru Dengue Fever Cases STL Decomp",
       subtitle = "July 2000 - June 2010")
```

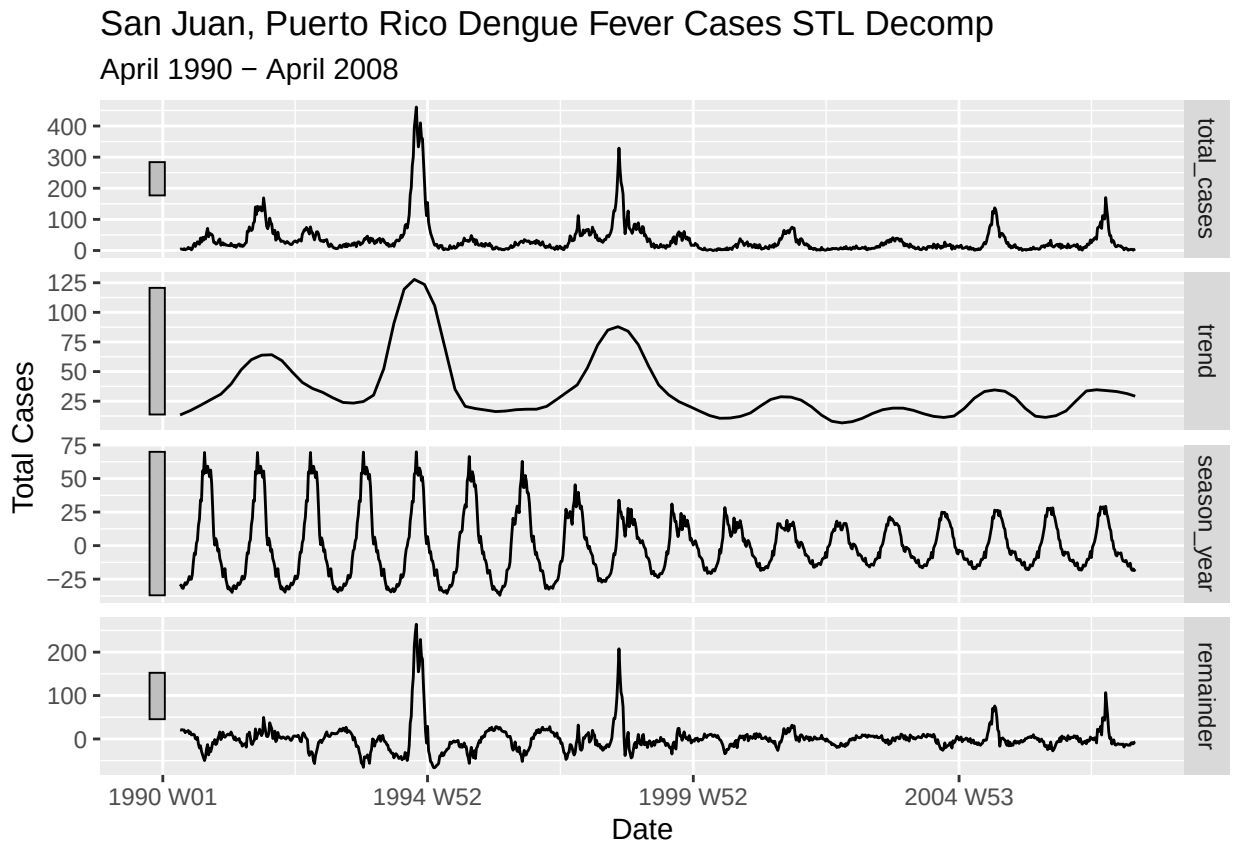



```
## Box-Cox
dengue_iq_train %>%
  model(STL(box_cox(total_cases, lambda))) %>%
  components() %>%
  autoplot() +
  labs(x = "Date", y = "Total Cases",
       title = "Iquitos, Peru Dengue Fever Cases STL Decomp (Box-Cox)",
       subtitle = "July 2000 - June 2010")
```

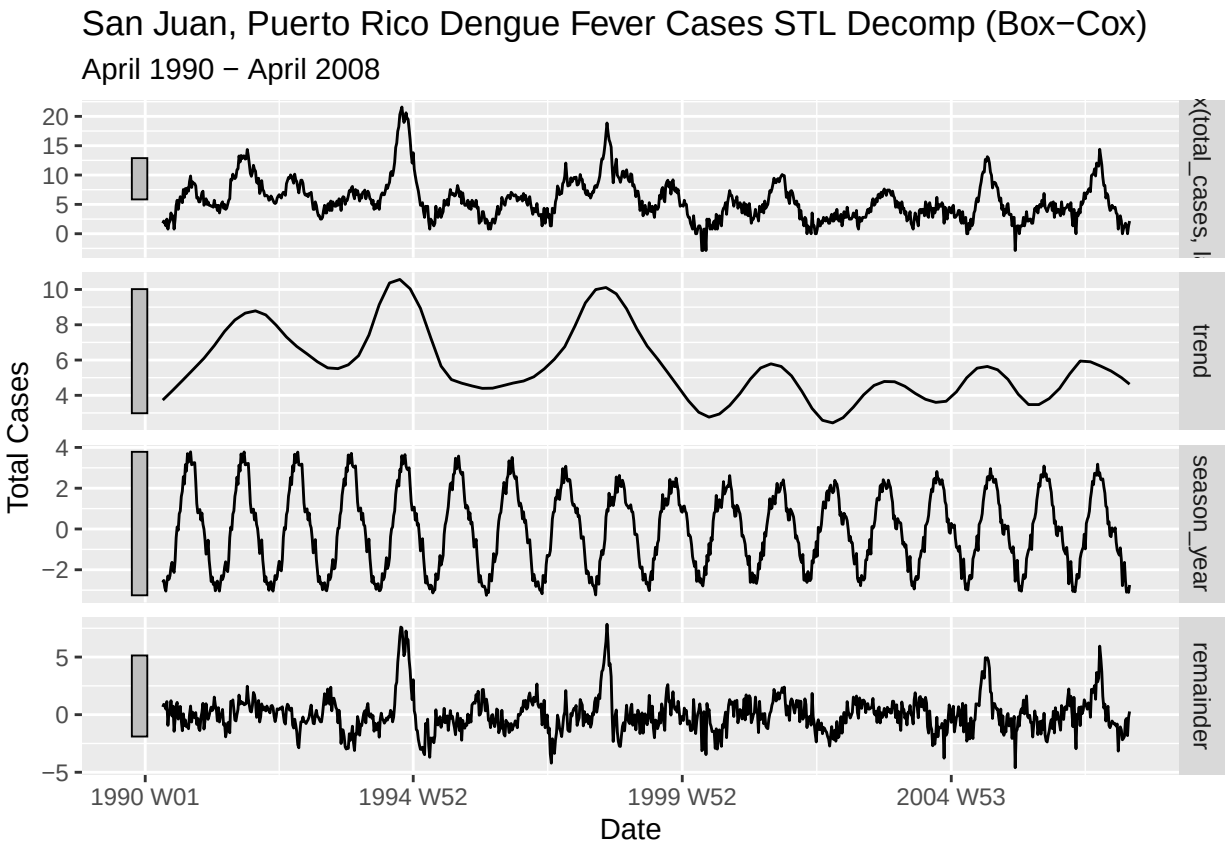


The data from Iquitos had consistent trend and seasonal effects, but due to seasonal variance in the data from San Juan I perform a Box-Cox transformation on the Iquitos data. As seen in the above plots, the data is still stable.

```
# Decompose San Juan
## Untransformed
dengue_sj_train %>%
  model(STL(total_cases)) %>%
  components() %>%
  autoplot() +
  labs(x = "Date", y = "Total Cases",
       title = "San Juan, Puerto Rico Dengue Fever Cases STL Decomp",
       subtitle = "April 1990 - April 2008")
```




```
## Box-Cox transformed
dengue_sj_train %>%
  model(STL(box_cox(total_cases, lambda))) %>%
  components() %>%
  autoplot() +
  labs(x = "Date", y = "Total Cases",
       title = "San Juan, Puerto Rico Dengue Fever Cases STL Decomp (Box-Cox)",
       subtitle = "April 1990 - April 2008")
```



Due to seasonal variance I transform the San Juan data using a Box-Cox transformation and $\lambda = 0.35$. While this does not perfectly stabilize the variance the data appears more stationary.

Modeling

I create three models with this data. An ARIMA model to act as a benchmark, using no predictors. Additionally, I create an ARIMA and a Neural Network using predictors, with some nudges on predictor selection and crafting based on previous literature. Each training set uses the same model to test for accuracy across different cities and determine applicability in other locations.

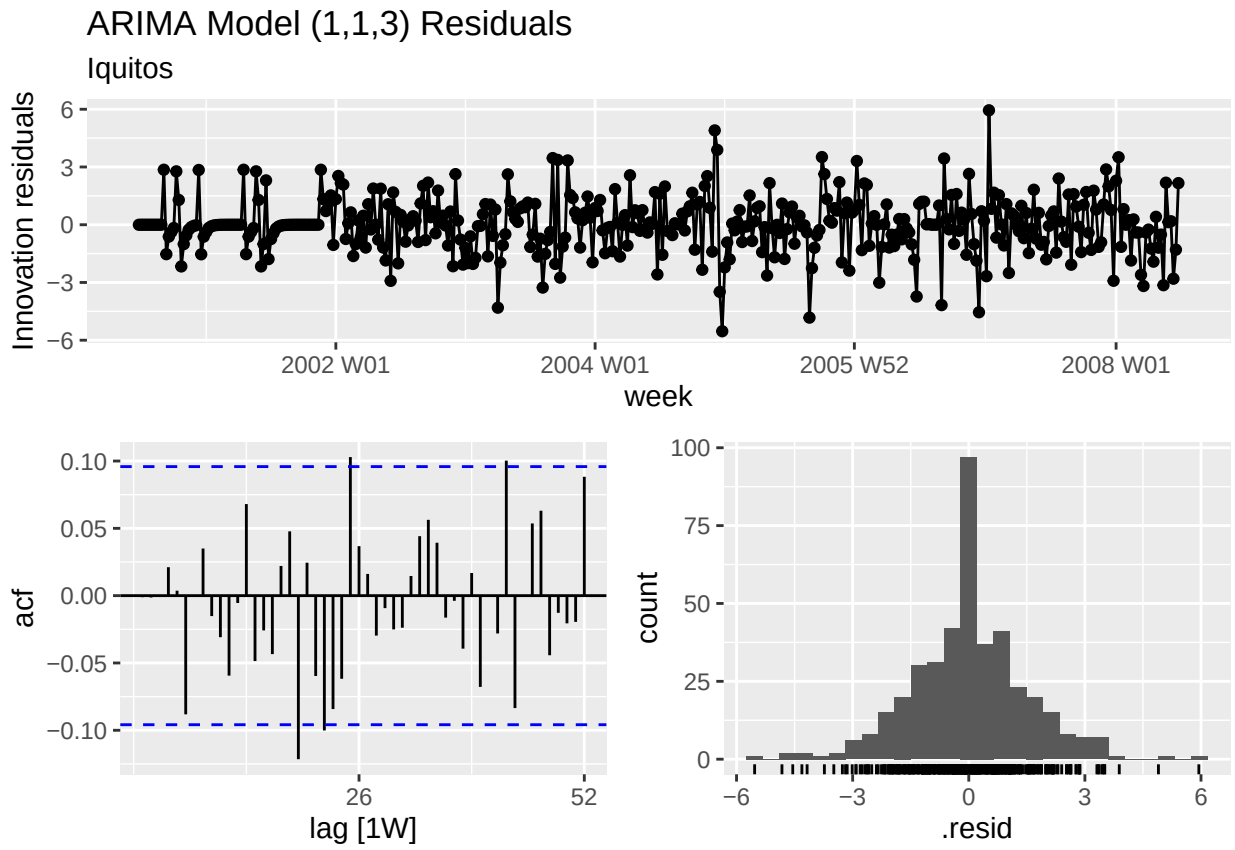
ARIMA (Benchmark Model)

```
# Iquitos
## Create Model
dengue_iq_arima = head(dengue_iq_train, round(train_iq_obs*0.8)) %>%
  model(ARIMA = ARIMA(box_cox(total_cases, lambda) ~ pdq(1,1,3) + PDQ(0,0,0)))

## Report
report(dengue_iq_arima)
```

```
## Series: total_cases
## Model: ARIMA(1,1,3)
## Transformation: box_cox(total_cases, lambda)
```

```
##
## Coefficients:
##           ar1      ma1      ma2     ma3
##       -0.0794  -0.3851  -0.0692  0.0124
## s.e.   1.8471   1.8463   0.8562  0.0783
##
## sigma^2 estimated as 2.211:  log likelihood=-755.25
## AIC=1520.5   AICc=1520.65   BIC=1540.67
```

```
## Plot Components and Residuals
dengue_iq_arima %>%
  gg_tsresiduals(lag_max = 52) +
  labs(title = "ARIMA Model (1,1,3) Residuals",
       subtitle = "Iquitos")
```



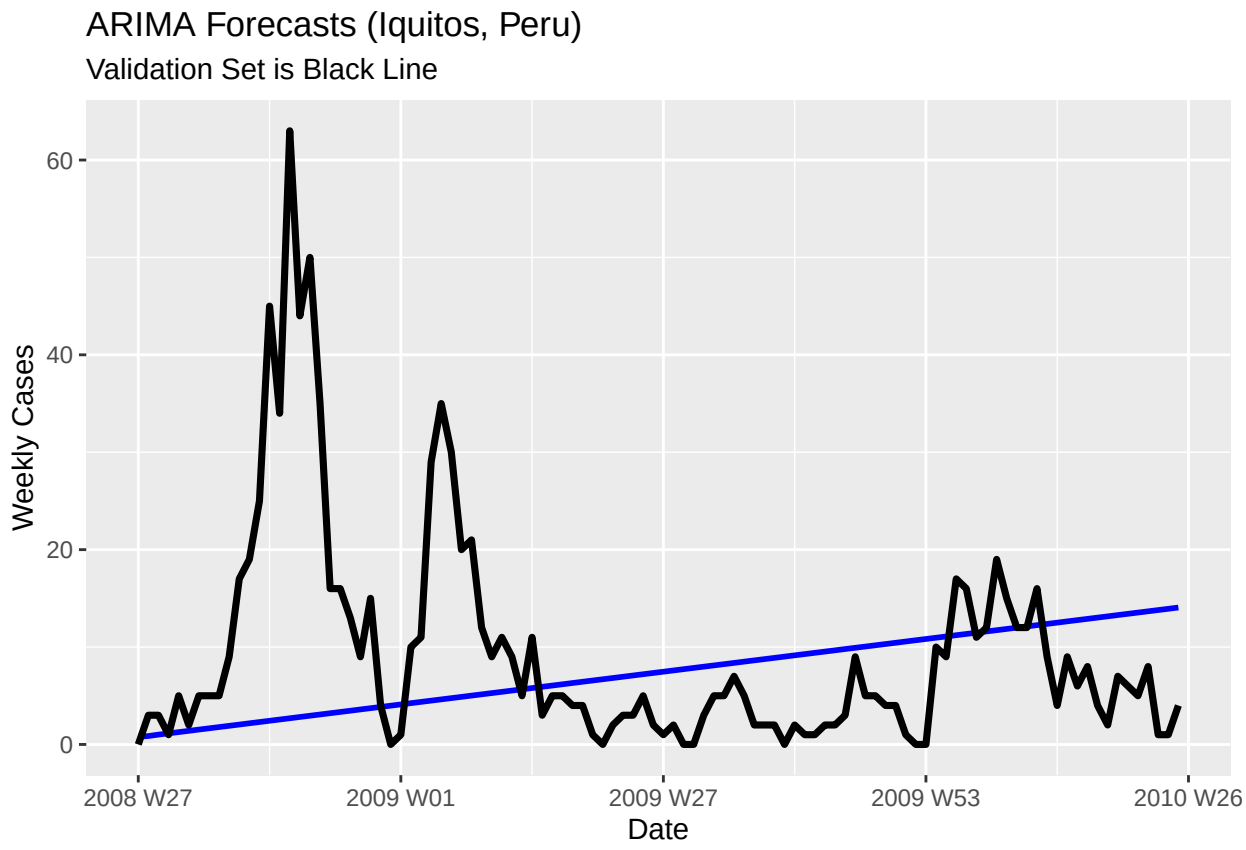

```
## Ljung-Box test
augment(dengue_iq_arima) %>%
  features(.innov, ljung_box, lag = 52)
```

```
## # A tibble: 1 x 4
##   city  .model lb_stat lb_pvalue
##   <chr> <chr>    <dbl>     <dbl>
## 1 iq    ARIMA     58.5     0.250
```

```r
## Accuracy of Model (focus on MAE since that is what the competition uses)
dengue_iq_arima_forecast = dengue_iq_arima %>%
  forecast(tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8)))

dengue_iq_arima_forecast %>%
  accuracy(tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8))) %>%
  select(.model, MAE)
```

```
## # A tibble: 1 x 2
##   .model   MAE
##   <chr>  <dbl>
## 1 ARIMA   8.65
```

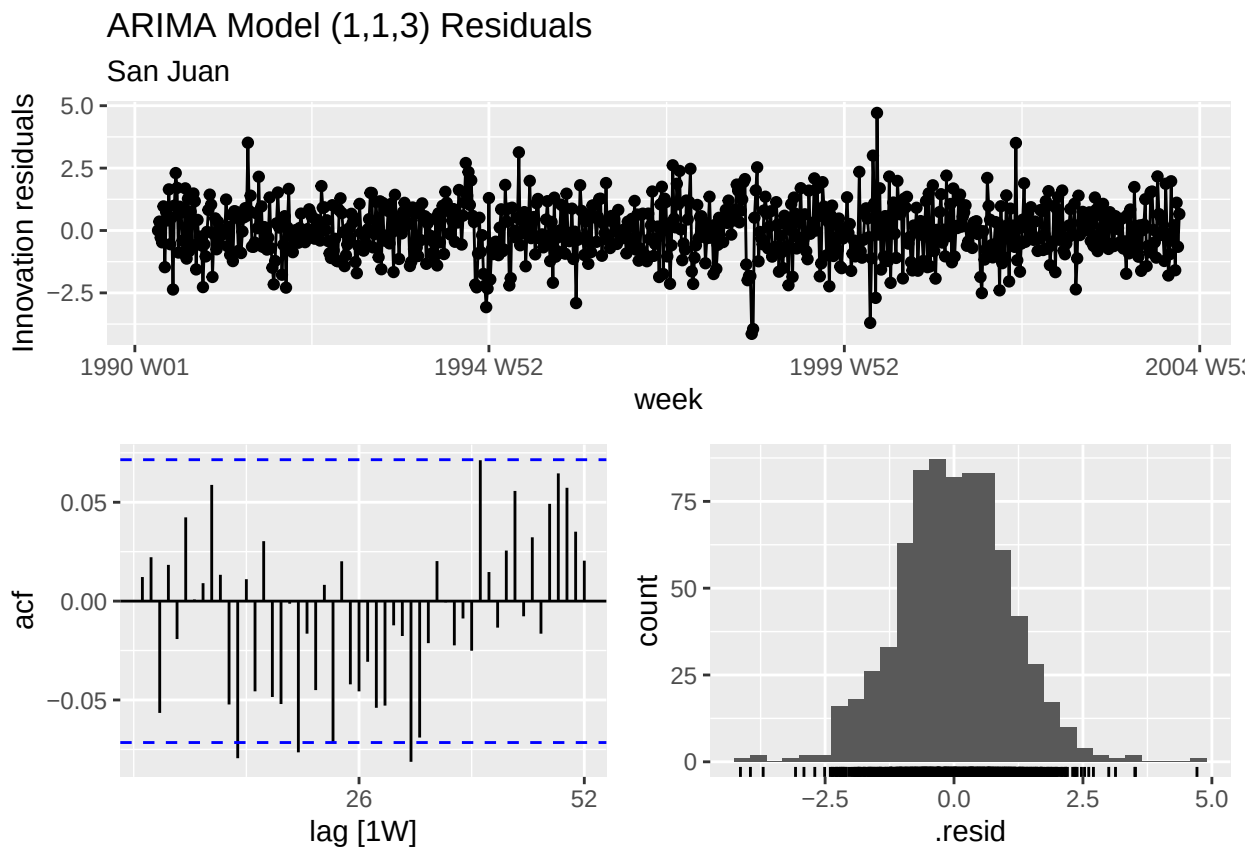
```r
## Plot Forecast
dengue_iq_arima_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "ARIMA Forecasts (Iquitos, Peru)",
       subtitle = "Validation Set is Black Line") +
  geom_line(data = tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8)),
            aes(x = week, y = total_cases),
            lwd = 1.25)
```



```
# San Juan
## Create Model
dengue_sj_arima = head(dengue_sj_train, round(train_sj_obs*0.8)) %>%
  model(ARIMA = ARIMA(box_cox(total_cases, lambda) ~ pdq(1,1,3) + PDQ(0,0,0)))

## Report
report(dengue_sj_arima)
```

```
## Series: total_cases
## Model: ARIMA(1,1,3)
## Transformation: box_cox(total_cases, lambda)
##
## Coefficients:
##          ar1      ma1     ma2     ma3
##       0.8134  -1.0903  0.1430  0.1669
## s.e.  0.0475   0.0562  0.0521  0.0369
##
## sigma^2 estimated as 1.18:  log likelihood=-1124.37
## AIC=2258.74   AICc=2258.82   BIC=2281.84
```

```
## Plot Components and Residuals
dengue_sj_arima %>%
  gg_tsresiduals(lag_max = 52) +
  labs(title = "ARIMA Model (1,1,3) Residuals",
       subtitle = "San Juan")
```



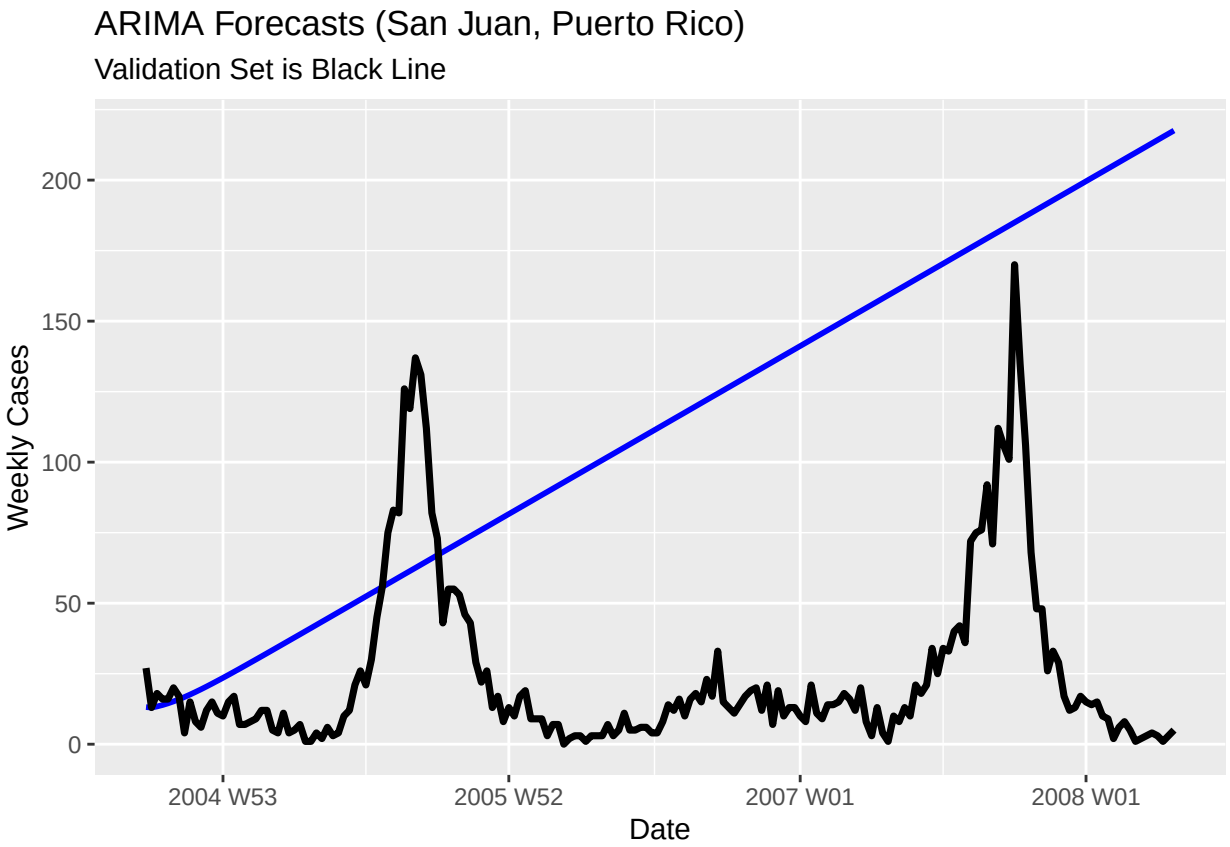
```
## Ljung-Box test
augment(dengue_sj_arima) %>%
  features(.innov, ljung_box, lag = 52)
```

```
## # A tibble: 1 x 4
##   city  .model lb_stat lb_pvalue
##   <chr> <chr>    <dbl>     <dbl>
## 1 sj    ARIMA     68.4    0.0631
```

```
## Accuracy of Model (focus on MAE since that is what the competition uses)
dengue_sj_arima_forecast = dengue_sj_arima %>%
  forecast(tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8)))

dengue_sj_arima_forecast %>%
  accuracy(tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8))) %>%
  select(.model, MAE)
```

```
## # A tibble: 1 x 2
##   .model   MAE
##   <chr>  <dbl>
## 1 ARIMA   92.6
```

```
## Plot Forecast
dengue_sj_arima_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "ARIMA Forecasts (San Juan, Puerto Rico)",
       subtitle = "Validation Set is Black Line") +
  geom_line(data = tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8)),
            aes(x = week, y = total_cases),
            lwd = 1.25)
```



I creat two non-seasonal ARIMA models choosing values of $p = 1, d = 1, q = 3$. The training data is split 80/20 for a train/validation set. The model fits the Iquitos data better than San Juan (AIC of 1938 and 2884 respectively). Plotting residuals from both sets of training data show potential issues with autocorrelation, but Ljung-Box tests on both sets show no significant problems. Accuracy measures on the validation data are good for Iquitos, but much greater for San Juan (MAE of 8.65 vs. 92.6). Plots of the forecast against the validation set show the models to have the appearance of a Drift model./

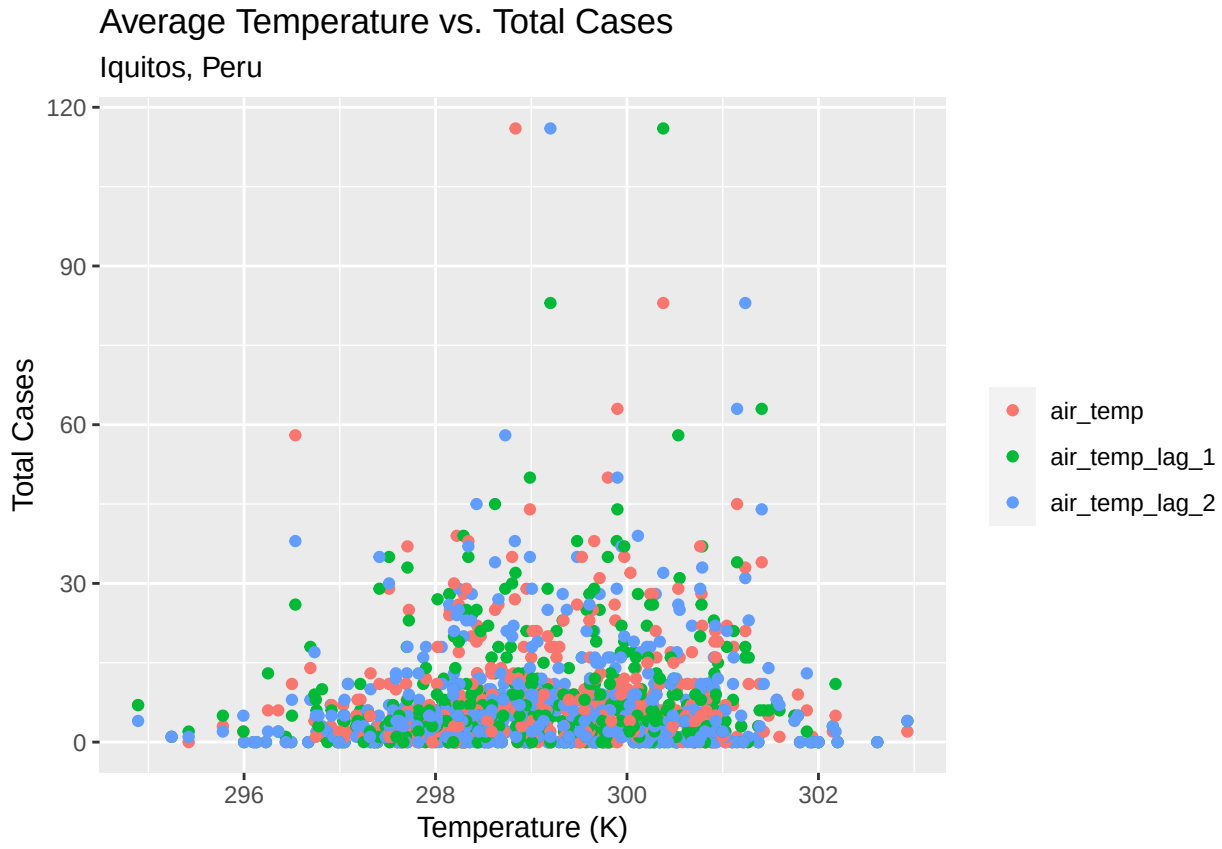
ARIMA with Predictors

Next, I continue with the ARIMA model and attempt to find a better fit and potential forecast by adding predictors and Fourier terms. Predictor selection is influenced by previous research on the topic.

Predictor Analysis Previous literature helped with early predictor selection or elimination. In the Kingdom of Saudi Arabia air temperature was found to be significantly associated with dengue fever, but humidity was not (Abualamah et al, 2021). I use temperature as my starting predictor and eliminate humidity from consideration. Additionally, clinical research into dengue fever can help shape choices to lag predictors. Dengue fever has an incubation period of 3-14 days with an average period of 7 days (Kularatne, 2015). I lag predictors by both a week and two weeks.

```
# Iquitos
## Plot
dengue_iq_train %>%
  select(total_cases, reanalysis_avg_temp_k) %>%
  mutate(air_temp = reanalysis_avg_temp_k) %>%
  mutate(air_temp_lag_1 = lag(reanalysis_avg_temp_k)) %>%
```

```
  mutate(air_temp_lag_2 = lag(reanalysis_avg_temp_k, 2)) %>%
  select(-reanalysis_avg_temp_k) %>%
  pivot_longer(cols = c(air_temp, air_temp_lag_1, air_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Temperature (K)", y = "Total Cases",
       title = "Average Temperature vs. Total Cases",
       subtitle = "Iquitos, Peru") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
temp_iq_cor_base = cor(dengue_iq_train$total_cases,
                       dengue_iq_train$reanalysis_avg_temp_k,
                       use = "complete.obs")

temp_iq_cor_lag_1 = cor(dengue_iq_train$total_cases,
                        lag(dengue_iq_train$reanalysis_avg_temp_k),
                        use = "complete.obs")

temp_iq_cor_lag_2 = cor(dengue_iq_train$total_cases,
                        lag(dengue_iq_train$reanalysis_avg_temp_k, 2),
                        use = "complete.obs")

temp_iq_cor = c(temp_iq_cor_base, temp_iq_cor_lag_1, temp_iq_cor_lag_2)
```

```r
cat("Iquitos, Peru Average Temperature Correlation\n")
```

```
## Iquitos, Peru Average Temperature Correlation
```

```r
print(matrix(data = temp_iq_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                  Base      Lag 1      Lag 2
## Correlation 0.0768732 0.07843576 0.08693761
```

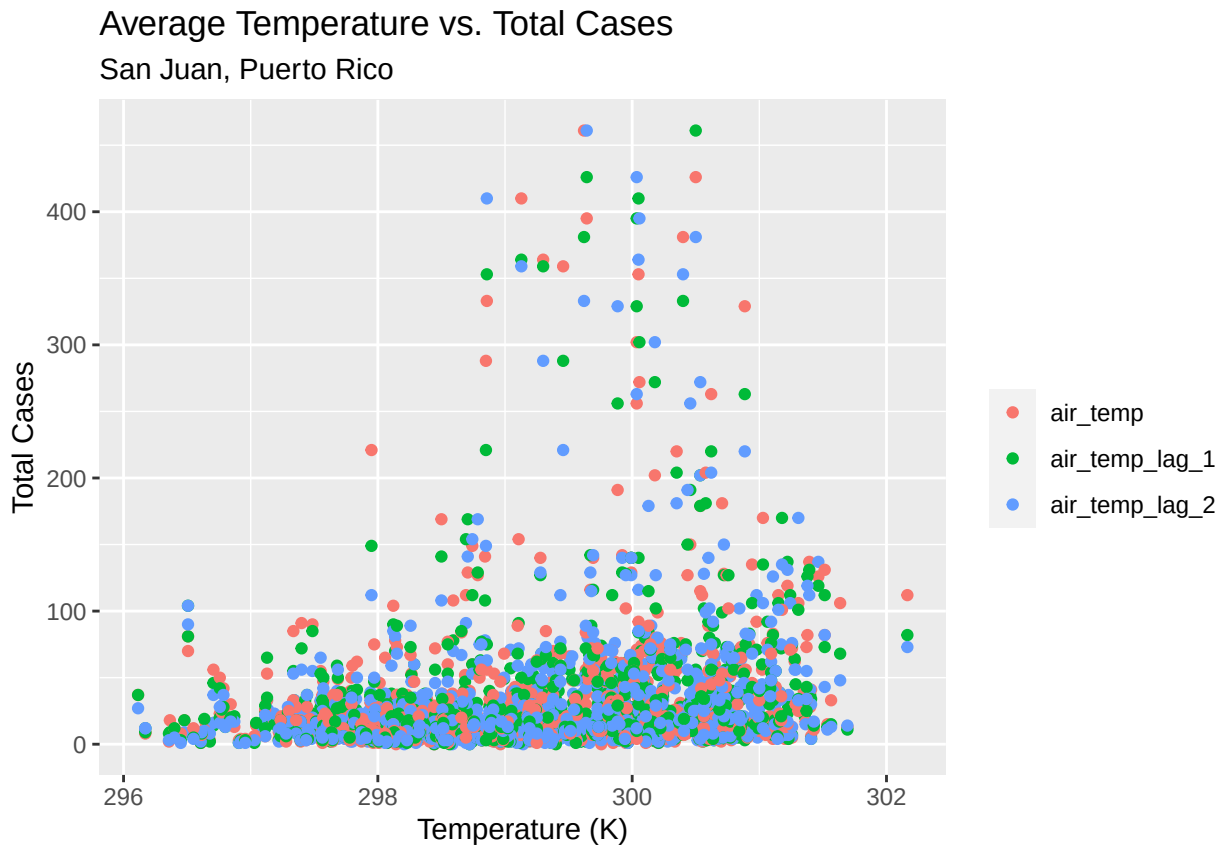
```r
## Stationarity Test
dengue_iq_train %>%
  features(reanalysis_avg_temp_k, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 iq       0.0653         0.1
```

```r
### San Juan was not stationary so I will difference both sets of data
dengue_iq_train %>%
  mutate(temp = difference(reanalysis_avg_temp_k)) %>%
  features(temp, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 iq      0.00964         0.1
```

```r
# San Juan
## Plot
dengue_sj_train %>%
  select(total_cases, reanalysis_avg_temp_k) %>%
  mutate(air_temp = reanalysis_avg_temp_k) %>%
  mutate(air_temp_lag_1 = lag(reanalysis_avg_temp_k)) %>%
  mutate(air_temp_lag_2 = lag(reanalysis_avg_temp_k, 2)) %>%
  select(-reanalysis_avg_temp_k) %>%
  pivot_longer(cols = c(air_temp, air_temp_lag_1, air_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Temperature (K)", y = "Total Cases",
       title = "Average Temperature vs. Total Cases",
       subtitle = "San Juan, Puerto Rico") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
temp_sj_cor_base = cor(dengue_sj_train$total_cases,
                       dengue_sj_train$reanalysis_avg_temp_k,
                       use = "complete.obs")

temp_sj_cor_lag_1 = cor(dengue_sj_train$total_cases,
                        lag(dengue_sj_train$reanalysis_avg_temp_k),
                        use = "complete.obs")

temp_sj_cor_lag_2 = cor(dengue_sj_train$total_cases,
                        lag(dengue_sj_train$reanalysis_avg_temp_k, 2),
                        use = "complete.obs")

temp_sj_cor = c(temp_sj_cor_base, temp_sj_cor_lag_1, temp_sj_cor_lag_2)

cat("San Juan, Puerto Rico Average Temperature Correlation\n")
```

```
## San Juan, Puerto Rico Average Temperature Correlation
```

```
print(matrix(data = temp_sj_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                  Base     Lag 1     Lag 2
## Correlation 0.1728141 0.1943173 0.2149463
```

```
## Stationarity Test
dengue_sj_train %>%
  features(reanalysis_avg_temp_k, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj        0.822        0.01
```

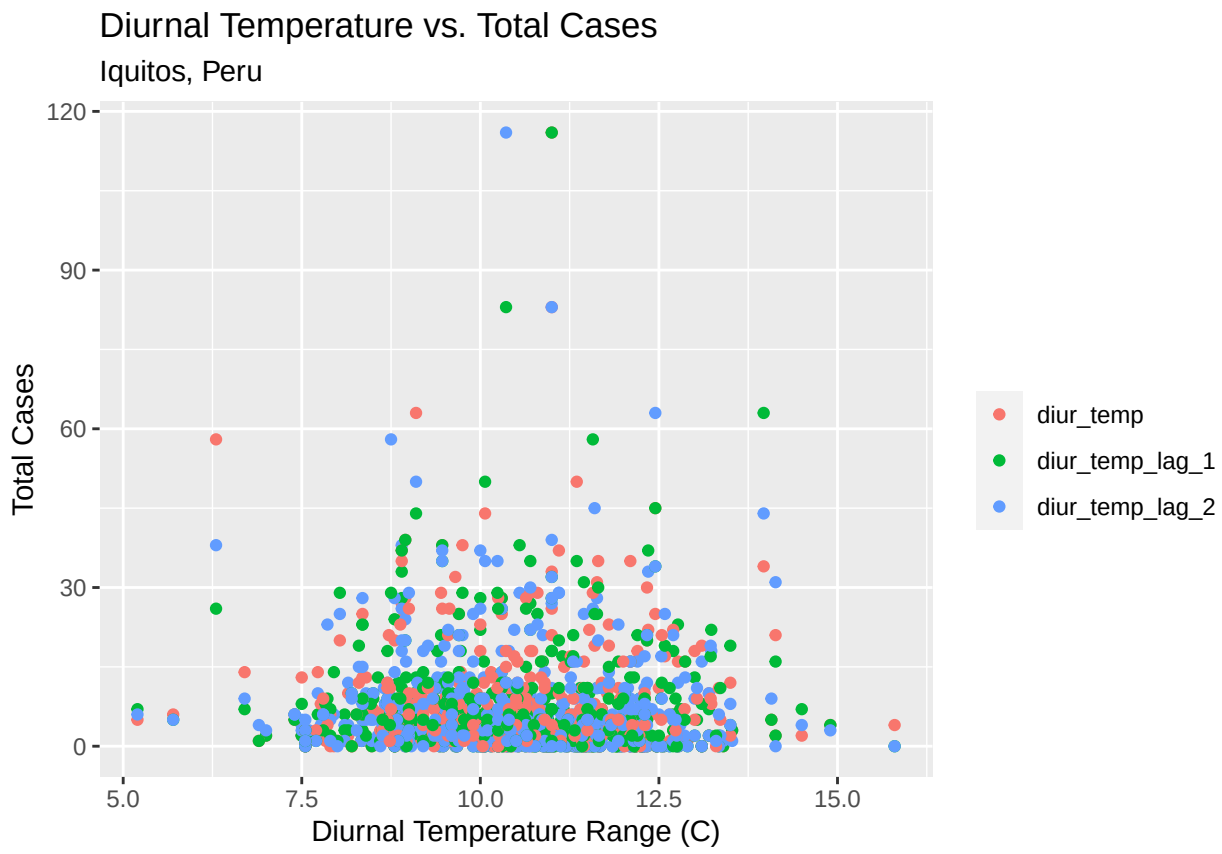
```
### Difference and test again
dengue_sj_train %>%
  mutate(temp = difference(reanalysis_avg_temp_k)) %>%
  features(temp, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj       0.0144         0.1
```

```
# Clean up values
rm(temp_iq_cor, temp_iq_cor_base, temp_iq_cor_lag_1, temp_iq_cor_lag_2,
   temp_sj_cor, temp_sj_cor_base, temp_sj_cor_lag_1, temp_sj_cor_lag_2)
```

This is low correlation between average temperature and cases per week, but the correlation gets stronger as values are lagged. There is greater correlation in San Juan. This is likely due to the higher number of cases recorded in that city. I include this predictor due to previous research towards its significance (Abualamah et al, 2021). The data from San Juan was not stationary so I difference both sets of data and checked again. Both sets of data are stationary when differenced.

```
# Iquitos
## Plot
dengue_iq_train %>%
  select(total_cases, station_diur_temp_rng_c) %>%
  mutate(diur_temp = station_diur_temp_rng_c) %>%
  mutate(diur_temp_lag_1 = lag(station_diur_temp_rng_c)) %>%
  mutate(diur_temp_lag_2 = lag(station_diur_temp_rng_c, 2)) %>%
  select(-station_diur_temp_rng_c) %>%
  pivot_longer(cols = c(diur_temp, diur_temp_lag_1, diur_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Diurnal Temperature Range (C)", y = "Total Cases",
       title = "Diurnal Temperature vs. Total Cases",
       subtitle = "Iquitos, Peru") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
diur_iq_cor_base = cor(dengue_iq_train$total_cases,
                       dengue_iq_train$station_diur_temp_rng_c,
                       use = "complete.obs")

diur_iq_cor_lag_1 = cor(dengue_iq_train$total_cases,
                        lag(dengue_iq_train$station_diur_temp_rng_c),
                        use = "complete.obs")

diur_iq_cor_lag_2 = cor(dengue_iq_train$total_cases,
                        lag(dengue_iq_train$station_diur_temp_rng_c, 2),
                        use = "complete.obs")

diur_iq_cor = c(diur_iq_cor_base, diur_iq_cor_lag_1, diur_iq_cor_lag_2)

cat("Iquitos, Peru Diurnal Temperature Correlation\n")
```

```
## Iquitos, Peru Diurnal Temperature Correlation
```

```
print(matrix(data = diur_iq_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                    Base        Lag 1      Lag 2
## Correlation -0.02148258 -0.002613993 -0.0297616
```

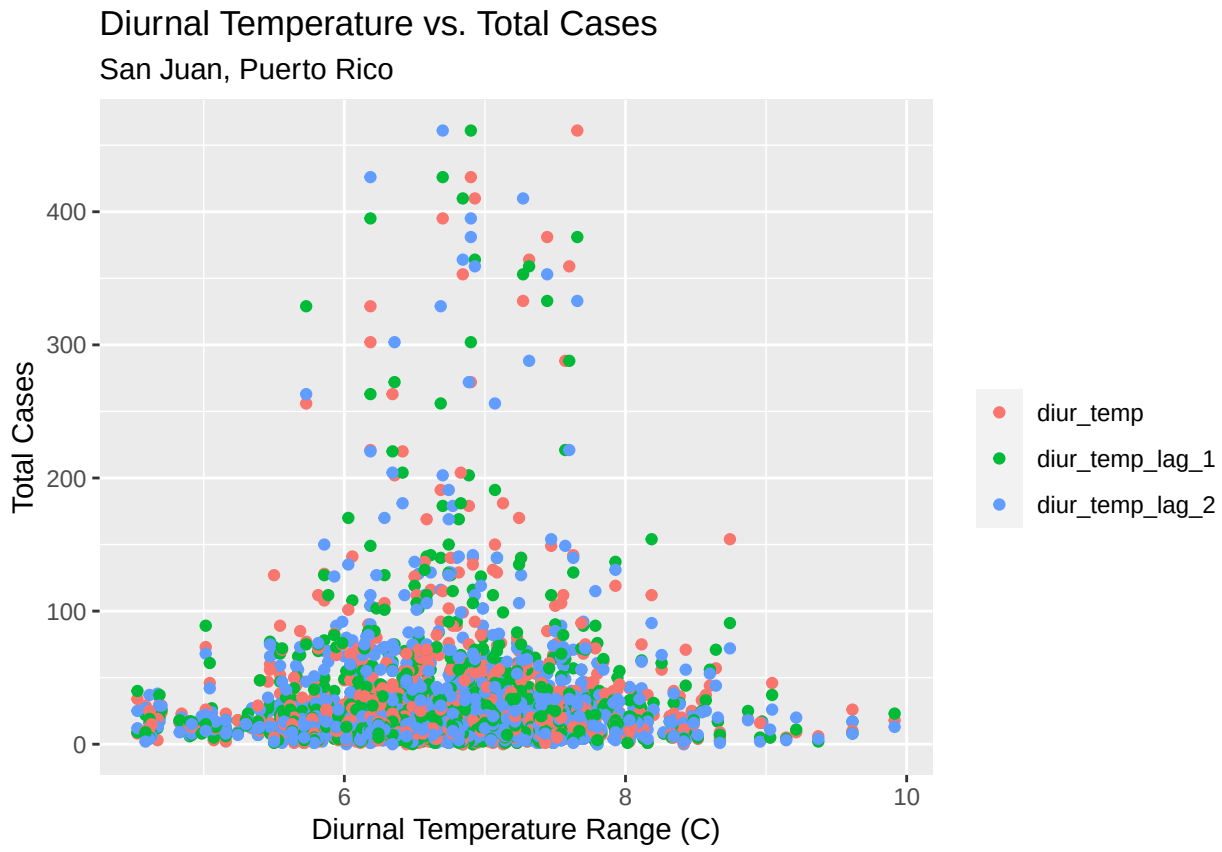
```r
## Stationarity
dengue_sj_train %>%
  features(station_diur_temp_rng_c, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj         2.37        0.01
```

```r
### Not stationary, difference and test again
dengue_sj_train %>%
  mutate(temp = difference(station_diur_temp_rng_c)) %>%
  features(temp, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj      0.00701         0.1
```

```r
# San Juan
## Plot
dengue_sj_train %>%
  select(total_cases, station_diur_temp_rng_c) %>%
  mutate(diur_temp = station_diur_temp_rng_c) %>%
  mutate(diur_temp_lag_1 = lag(station_diur_temp_rng_c)) %>%
  mutate(diur_temp_lag_2 = lag(station_diur_temp_rng_c, 2)) %>%
  select(-station_diur_temp_rng_c) %>%
  pivot_longer(cols = c(diur_temp, diur_temp_lag_1, diur_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Diurnal Temperature Range (C)", y = "Total Cases",
       title = "Diurnal Temperature vs. Total Cases",
       subtitle = "San Juan, Puerto Rico") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
diur_sj_cor_base = cor(dengue_sj_train$total_cases,
                       dengue_sj_train$station_diur_temp_rng_c,
                       use = "complete.obs")

diur_sj_cor_lag_1 = cor(dengue_sj_train$total_cases,
                        lag(dengue_sj_train$station_diur_temp_rng_c),
                        use = "complete.obs")

diur_sj_cor_lag_2 = cor(dengue_sj_train$total_cases,
                        lag(dengue_sj_train$station_diur_temp_rng_c, 2),
                        use = "complete.obs")

diur_sj_cor = c(diur_sj_cor_base, diur_sj_cor_lag_1, diur_sj_cor_lag_2)

cat("San Juan, Puerto Rico Diurnal Temperature Correlation\n")
```

```
## San Juan, Puerto Rico Diurnal Temperature Correlation
```

```
print(matrix(data = diur_sj_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                    Base      Lag 1      Lag 2
## Correlation 0.03578021 0.01910245 0.01182082
```

```
## Stationarity
dengue_sj_train %>%
  features(station_diur_temp_rng_c, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj         2.37        0.01
```

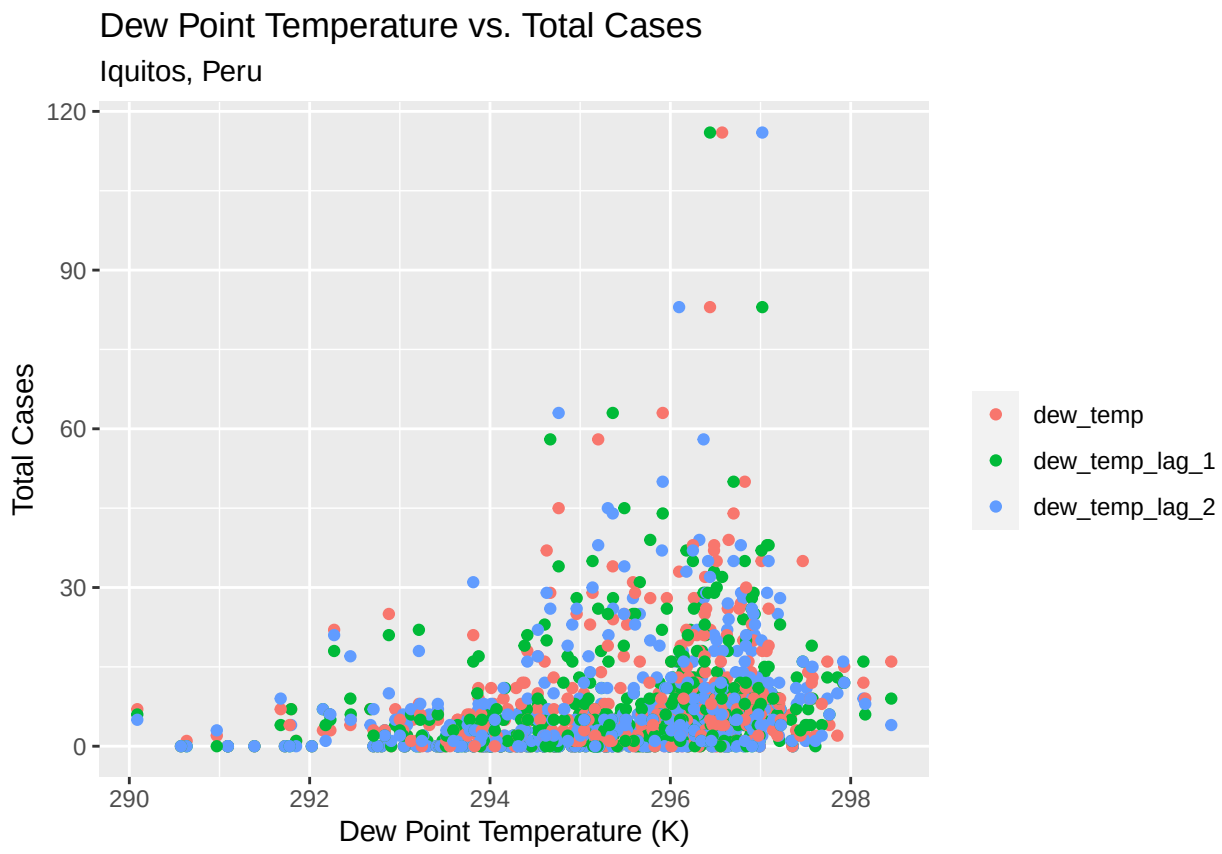
```
### Iquitos was not stationary, difference and retest
dengue_sj_train %>%
  mutate(temp = difference(station_diur_temp_rng_c)) %>%
  features(temp, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj      0.00701         0.1
```

```
# Clean up values
rm(diur_iq_cor, diur_iq_cor_base, diur_iq_cor_lag_1, diur_iq_cor_lag_2,
   diur_sj_cor, diur_sj_cor_base, diur_sj_cor_lag_1, diur_sj_cor_lag_2)
```

There is negative correlation from the diurnal temperature in Iquitos, but positive correlation in San Juan. The data is stationary after differencing, given the opposite correlation I do not use this predictor in the modeling.

```
# Iquitos
## Plot
dengue_iq_train %>%
  select(total_cases, reanalysis_dew_point_temp_k) %>%
  mutate(dew_temp = reanalysis_dew_point_temp_k) %>%
  mutate(dew_temp_lag_1 = lag(reanalysis_dew_point_temp_k)) %>%
  mutate(dew_temp_lag_2 = lag(reanalysis_dew_point_temp_k, 2)) %>%
  select(-reanalysis_dew_point_temp_k) %>%
  pivot_longer(cols = c(dew_temp, dew_temp_lag_1, dew_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Dew Point Temperature (K)", y = "Total Cases",
       title = "Dew Point Temperature vs. Total Cases",
       subtitle = "Iquitos, Peru") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
dew_iq_cor_base = cor(dengue_iq_train$total_cases,
                      dengue_iq_train$reanalysis_dew_point_temp_k,
                      use = "complete.obs")

dew_iq_cor_lag_1 = cor(dengue_iq_train$total_cases,
                       lag(dengue_iq_train$reanalysis_dew_point_temp_k),
                       use = "complete.obs")

dew_iq_cor_lag_2 = cor(dengue_iq_train$total_cases,
                       lag(dengue_iq_train$reanalysis_dew_point_temp_k, 2),
                       use = "complete.obs")

dew_iq_cor = c(dew_iq_cor_base, dew_iq_cor_lag_1, dew_iq_cor_lag_2)

cat("Iquitos, Peru Dew Point Temperature Correlation\n")
```

```
## Iquitos, Peru Dew Point Temperature Correlation
```

```
print(matrix(data = dew_iq_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                  Base     Lag 1     Lag 2
## Correlation 0.2295598 0.2220186 0.2156592
```

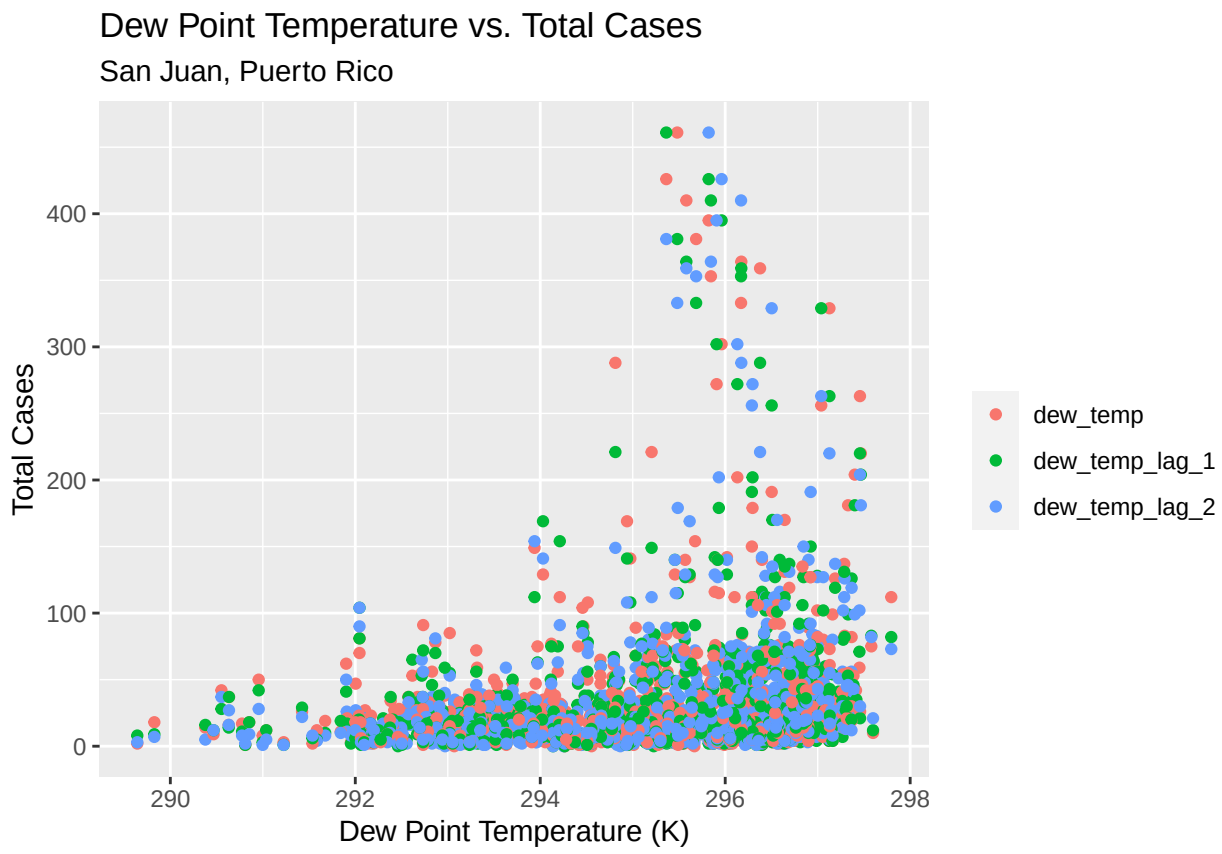
```r
## Stationarity test
dengue_iq_train %>%
  features(reanalysis_dew_point_temp_k, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 iq         1.10        0.01
```

```r
### Not stationary, difference and retest
dengue_iq_train %>%
  mutate(temp = difference(reanalysis_dew_point_temp_k)) %>%
  features(temp, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 iq      0.00954         0.1
```

```r
# San Juan
## Plot
dengue_sj_train %>%
  select(total_cases, reanalysis_dew_point_temp_k) %>%
  mutate(dew_temp = reanalysis_dew_point_temp_k) %>%
  mutate(dew_temp_lag_1 = lag(reanalysis_dew_point_temp_k)) %>%
  mutate(dew_temp_lag_2 = lag(reanalysis_dew_point_temp_k, 2)) %>%
  select(-reanalysis_dew_point_temp_k) %>%
  pivot_longer(cols = c(dew_temp, dew_temp_lag_1, dew_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Dew Point Temperature (K)", y = "Total Cases",
       title = "Dew Point Temperature vs. Total Cases",
       subtitle = "San Juan, Puerto Rico") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
dew_sj_cor_base = cor(dengue_sj_train$total_cases,
                      dengue_sj_train$reanalysis_dew_point_temp_k,
                      use = "complete.obs")

dew_sj_cor_lag_1 = cor(dengue_sj_train$total_cases,
                       lag(dengue_sj_train$reanalysis_dew_point_temp_k),
                       use = "complete.obs")

dew_sj_cor_lag_2 = cor(dengue_sj_train$total_cases,
                       lag(dengue_sj_train$reanalysis_dew_point_temp_k, 2),
                       use = "complete.obs")

dew_sj_cor = c(dew_sj_cor_base, dew_sj_cor_lag_1, dew_sj_cor_lag_2)

cat("San Juan, Puerto Rico Dew Point Temperature Correlation\n")
```

```
## San Juan, Puerto Rico Dew Point Temperature Correlation
```

```
print(matrix(data = dew_sj_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                  Base     Lag 1     Lag 2
## Correlation 0.2015074 0.2223002 0.2442758
```

```
## Stationarity
dengue_sj_train %>%
  features(reanalysis_dew_point_temp_k, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj       0.0802         0.1
```

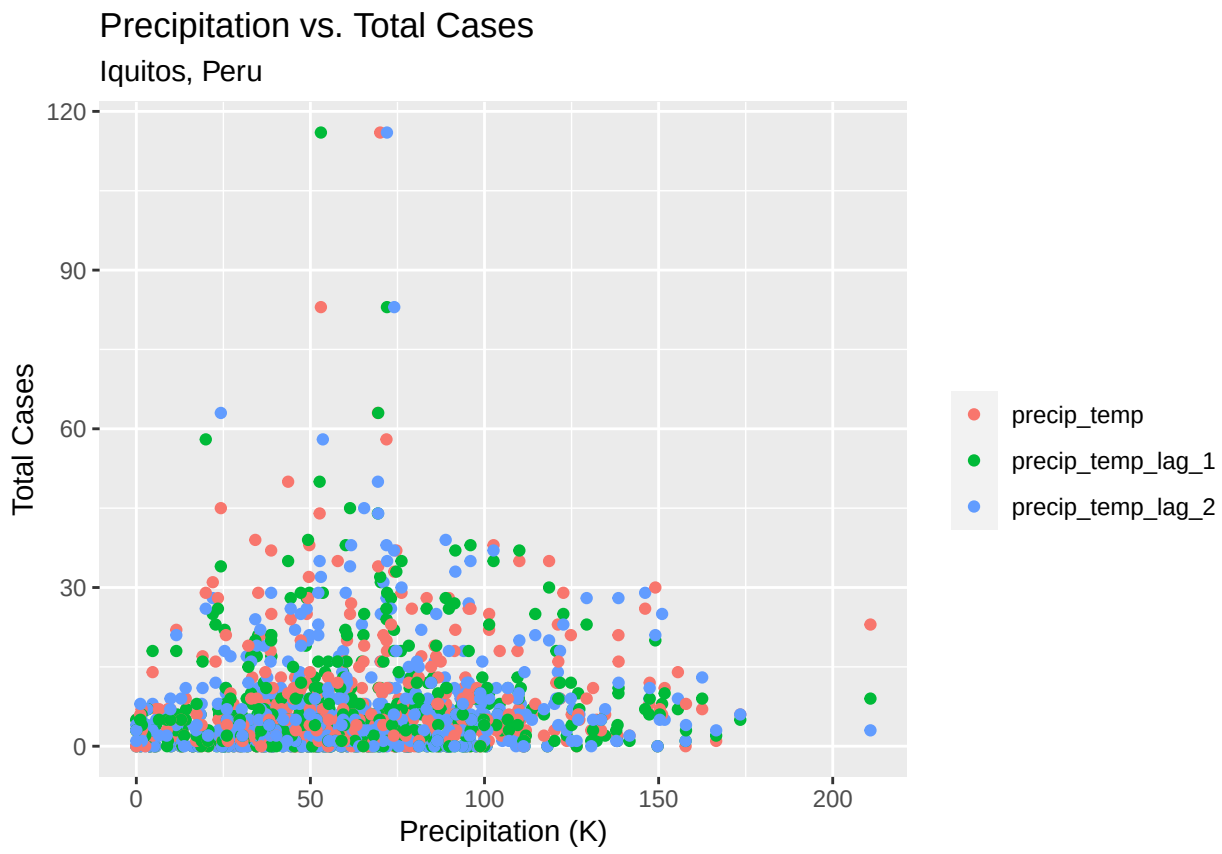
```
### Iquitos data not stationary, difference and retest
dengue_sj_train %>%
  mutate(temp = difference(reanalysis_dew_point_temp_k)) %>%
  features(temp, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj       0.0176         0.1
```

```
# Clean up values
rm(dew_iq_cor, dew_iq_cor_base, dew_iq_cor_lag_1, dew_iq_cor_lag_2,
   dew_sj_cor, dew_sj_cor_base, dew_sj_cor_lag_1, dew_sj_cor_lag_2)
```

Dew point temperature shows the strongest correlation of the available predictors, even while still being weak. There is increased correlation as values are lagged in San Juan, but the correlation stays consistent in Iquitos regardless of lag time. Data are not stationary in Iquitos, but differencing both sets of data solves the issue as with other predictors.

```
# Iquitos
## Plot
dengue_iq_train %>%
  select(total_cases, reanalysis_sat_precip_amt_mm) %>%
  mutate(precip_temp = reanalysis_sat_precip_amt_mm) %>%
  mutate(precip_temp_lag_1 = lag(reanalysis_sat_precip_amt_mm)) %>%
  mutate(precip_temp_lag_2 = lag(reanalysis_sat_precip_amt_mm, 2)) %>%
  select(-reanalysis_sat_precip_amt_mm) %>%
  pivot_longer(cols = c(precip_temp, precip_temp_lag_1, precip_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Precipitation (K)", y = "Total Cases",
       title = "Precipitation vs. Total Cases",
       subtitle = "Iquitos, Peru") +
  guides(color = guide_legend(title = ""))
```




```
## Correlation
precip_iq_cor_base = cor(dengue_iq_train$total_cases,
                         dengue_iq_train$reanalysis_sat_precip_amt_mm,
                         use = "complete.obs")

precip_iq_cor_lag_1 = cor(dengue_iq_train$total_cases,
                          lag(dengue_iq_train$reanalysis_sat_precip_amt_mm),
                          use = "complete.obs")

precip_iq_cor_lag_2 = cor(dengue_iq_train$total_cases,
                          lag(dengue_iq_train$reanalysis_sat_precip_amt_mm, 2),
                          use = "complete.obs")

precip_iq_cor = c(precip_iq_cor_base, precip_iq_cor_lag_1, precip_iq_cor_lag_2)

cat("Iquitos, Peru Precipitation Correlation\n")
```

```
## Iquitos, Peru Precipitation Correlation
```

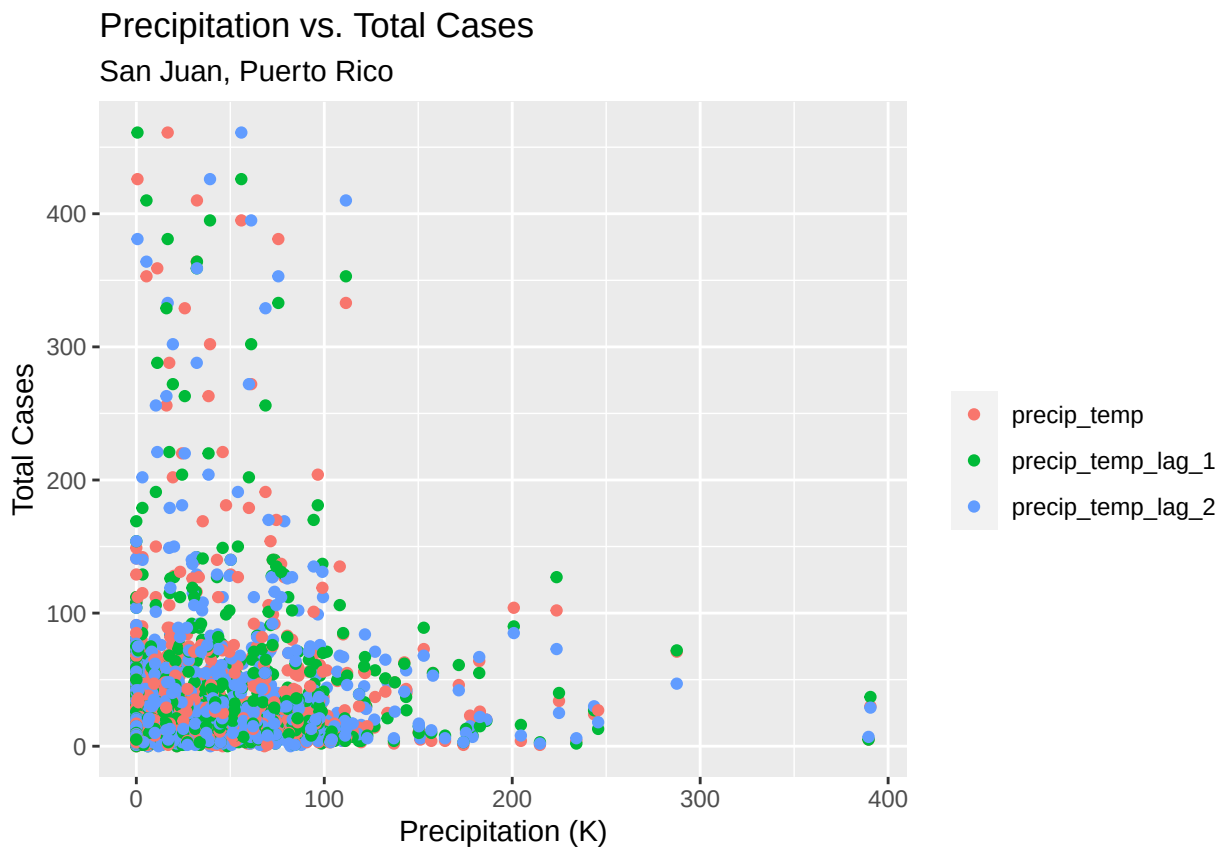
```
print(matrix(data = precip_iq_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                   Base      Lag 1      Lag 2
## Correlation 0.08967732 0.05764502 0.09011901
```

```
## Stationarity
dengue_iq_train %>%
  features(reanalysis_sat_precip_amt_mm, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 iq        0.248         0.1
```

```
# San Juan
## Plot
dengue_sj_train %>%
  select(total_cases, reanalysis_sat_precip_amt_mm) %>%
  mutate(precip_temp = reanalysis_sat_precip_amt_mm) %>%
  mutate(precip_temp_lag_1 = lag(reanalysis_sat_precip_amt_mm)) %>%
  mutate(precip_temp_lag_2 = lag(reanalysis_sat_precip_amt_mm, 2)) %>%
  select(-reanalysis_sat_precip_amt_mm) %>%
  pivot_longer(cols = c(precip_temp, precip_temp_lag_1, precip_temp_lag_2),
               names_to = "type", values_to = "value") %>%
  ggplot(aes(x = value, y = total_cases, color = type)) +
  geom_point() +
  labs(x = "Precipitation (K)", y = "Total Cases",
       title = "Precipitation vs. Total Cases",
       subtitle = "San Juan, Puerto Rico") +
  guides(color = guide_legend(title = ""))
```



```
## Correlation
precip_sj_cor_base = cor(dengue_sj_train$total_cases,
                         dengue_sj_train$reanalysis_sat_precip_amt_mm,
                         use = "complete.obs")

precip_sj_cor_lag_1 = cor(dengue_sj_train$total_cases,
                          lag(dengue_sj_train$reanalysis_sat_precip_amt_mm),
                          use = "complete.obs")

precip_sj_cor_lag_2 = cor(dengue_sj_train$total_cases,
                          lag(dengue_sj_train$reanalysis_sat_precip_amt_mm, 2),
                          use = "complete.obs")

precip_sj_cor = c(precip_sj_cor_base, precip_sj_cor_lag_1, precip_sj_cor_lag_2)

cat("San Juan, Puerto Rico Precipitation Correlation\n")
```

```
## San Juan, Puerto Rico Precipitation Correlation
```

```
print(matrix(data = precip_sj_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                   Base      Lag 1      Lag 2
## Correlation 0.05770377 0.07396006 0.08099985
```

```
## Stationarity
dengue_sj_train %>%
  features(reanalysis_sat_precip_amt_mm, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj        0.219         0.1
```

```
# Clean up values
rm(precip_iq_cor, precip_iq_cor_base, precip_iq_cor_lag_1, precip_iq_cor_lag_2,
   precip_sj_cor, precip_sj_cor_base, precip_sj_cor_lag_1, precip_sj_cor_lag_2)
```

Precipitation shows weak correlation, but with slight increases as values are lagged. Since there is consistency across both cities I include the predictor. Additionally, the data are stationary across both cities. With that, the three predictors considered in the ARIMA are average temperature, dew point temperature, and precipitation.

```
dengue_iq_train = dengue_iq_train %>%
  mutate(temp_diff = difference(reanalysis_avg_temp_k),
         dew_diff = difference(reanalysis_dew_point_temp_k),
         precip_diff = difference(reanalysis_sat_precip_amt_mm)) %>%
  mutate(temp_diff_lag1 = lag(temp_diff),
         temp_diff_lag2 = lag(temp_diff, 2),
         dew_diff_lag1 = lag(dew_diff),
         dew_diff_lag2 = lag(dew_diff, 2),
```

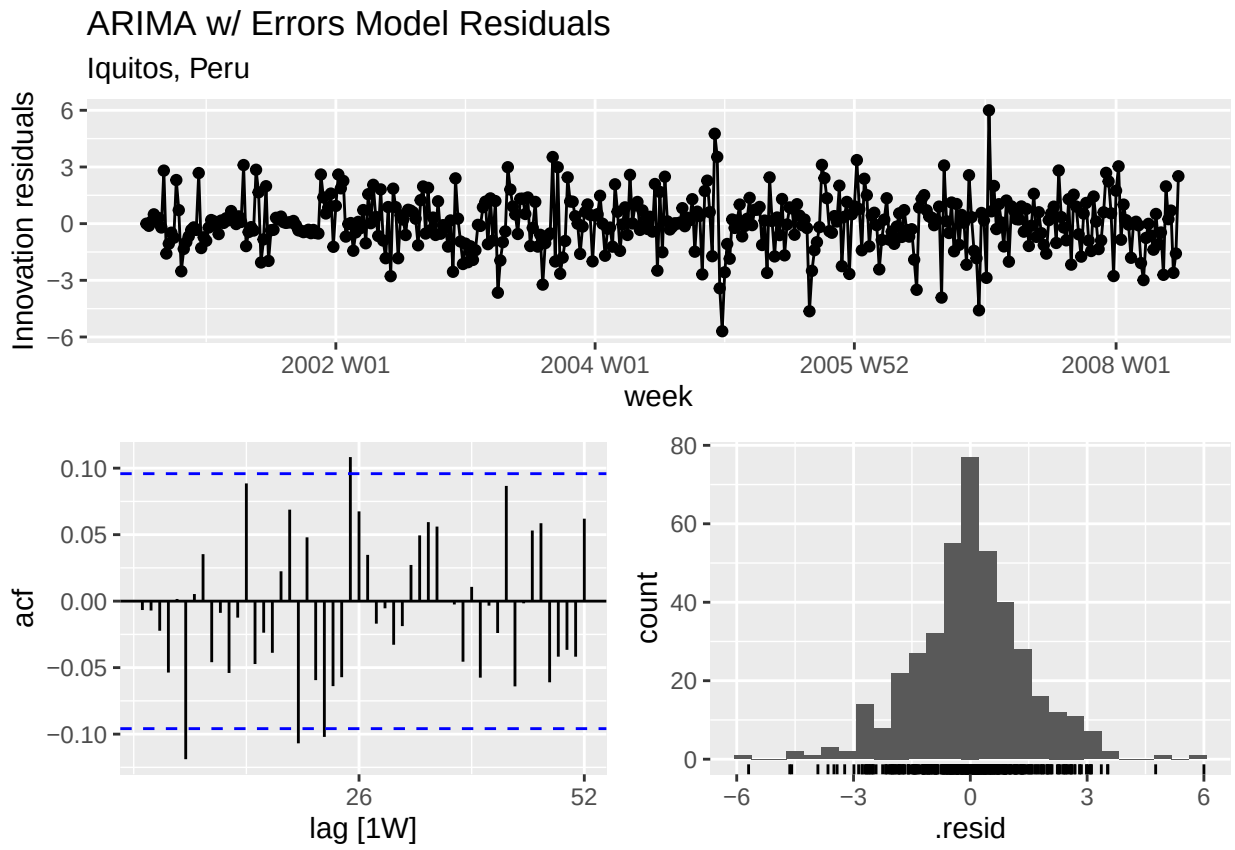
```
         precip_diff_lag1 = lag(precip_diff),
         precip_diff_lag2 = lag(precip_diff, 2))

dengue_sj_train = dengue_sj_train %>%
  mutate(temp_diff = difference(reanalysis_avg_temp_k),
         dew_diff = difference(reanalysis_dew_point_temp_k),
         precip_diff = difference(reanalysis_sat_precip_amt_mm)) %>%
  mutate(temp_diff_lag1 = lag(temp_diff),
         temp_diff_lag2 = lag(temp_diff, 2),
         dew_diff_lag1 = lag(dew_diff),
         dew_diff_lag2 = lag(dew_diff, 2),
         precip_diff_lag1 = lag(precip_diff),
         precip_diff_lag2 = lag(precip_diff, 2))
```

I adjust the data to create variables representing the difference of the selected predictors. In the following models those are lagged one and two weeks respectively to create the model.

```
# Iquitos
## Model
dengue_iq_arima2 = head(dengue_iq_train, round(train_iq_obs*0.8)) %>%
  model(
    `ARIMA w/ Errors` = ARIMA(box_cox(total_cases, lambda) ~ temp_diff_lag1 +
                                temp_diff_lag2 + dew_diff_lag1 + dew_diff_lag2 +
                                precip_diff_lag1 + precip_diff_lag2 +
                                pdq(3,1,0) + PDQ(0,0,0) + fourier(K = 3))
  )

## Report
report(dengue_iq_arima2)
```

```
## Series: total_cases
## Model: LM w/ ARIMA(3,1,0) errors
## Transformation: box_cox(total_cases, lambda)
##
## Coefficients:
##           ar1      ar2      ar3  temp_diff_lag1  temp_diff_lag2  dew_diff_lag1
##       -0.4882  -0.2643  -0.1123         -0.0071         -0.0513         0.0174
## s.e.   0.0495   0.0538   0.0497          0.0528          0.0520         0.0596
##       dew_diff_lag2  precip_diff_lag1  precip_diff_lag2  fourier(K = 3)C1_52
##             -0.0154           -0.0028           -0.0036               0.4921
## s.e.         0.0607            0.0016            0.0016               0.4531
##       fourier(K = 3)S1_52  fourier(K = 3)C2_52  fourier(K = 3)S2_52
##                    1.0852               0.1989               0.1670
## s.e.               0.4526               0.2322               0.2308
##       fourier(K = 3)C3_52  fourier(K = 3)S3_52
##                    0.0646              -0.1741
## s.e.               0.1591               0.1595
##
## sigma^2 estimated as 2.179:  log likelihood=-742.75
## AIC=1517.49   AICc=1518.85   BIC=1582.02
```

```
## Plot Residuals
dengue_iq_arima2 %>%
  gg_tsresiduals(lag_max = 52) +
  labs(title = "ARIMA w/ Errors Model Residuals",
       subtitle = "Iquitos, Peru")
```




```
## Ljung-Box Test
dengue_iq_arima2 %>%
  augment() %>%
  features(.innov, ljung_box, lag = 52)
```

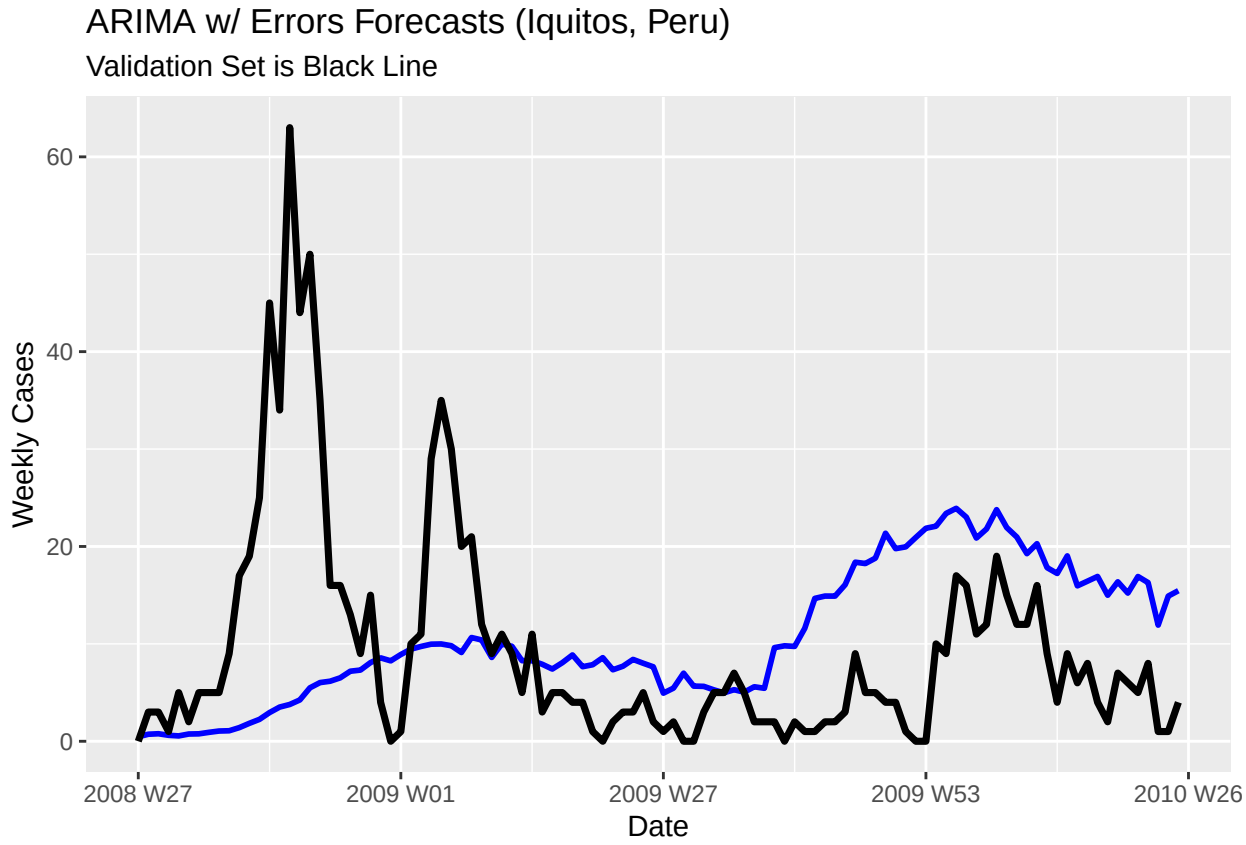
```
## # A tibble: 1 x 4
##   city  .model          lb_stat lb_pvalue
##   <chr> <chr>             <dbl>     <dbl>
## 1 iq    ARIMA w/ Errors    63.2     0.137
```

```
## Accuracy of Model (focus on MAE since that is what the competition uses)
dengue_iq_arima2_forecast = dengue_iq_arima2 %>%
  forecast(tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8)))

dengue_iq_arima2_forecast %>%
  accuracy(tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8))) %>%
  select(.model, MAE)
```

```
## # A tibble: 1 x 2
```

```
##   .model            MAE
##   <chr>           <dbl>
## 1 ARIMA w/ Errors  9.97
```

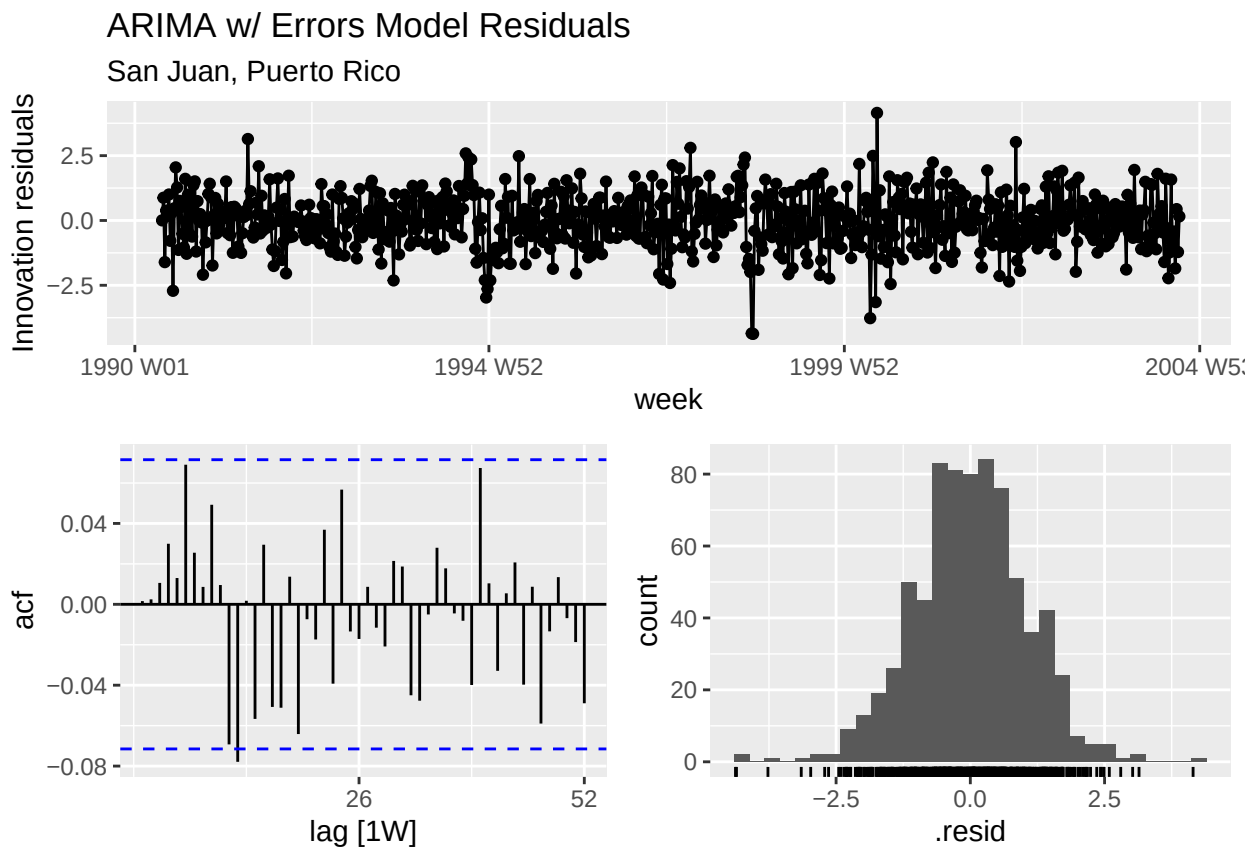
```
## Plot Forecast
dengue_iq_arima2_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "ARIMA w/ Errors Forecasts (Iquitos, Peru)",
       subtitle = "Validation Set is Black Line") +
  geom_line(data = tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8)),
            aes(x = week, y = total_cases),
            lwd = 1.25)
```




```
# San Juan
## Model
dengue_sj_arima2 = head(dengue_sj_train, round(train_sj_obs*0.8)) %>%
  model(
    `ARIMA w/ Errors` = ARIMA(box_cox(total_cases, lambda) ~ temp_diff_lag1 +
                                temp_diff_lag2 + dew_diff_lag1 + dew_diff_lag2 +
                                precip_diff_lag1 + precip_diff_lag2 +
                                pdq(3,1,0) + PDQ(0,0,0) + fourier(K = 3))
  )

## Report
report(dengue_sj_arima2)
```

```
## Series: total_cases
## Model: LM w/ ARIMA(3,1,0) errors
## Transformation: box_cox(total_cases, lambda)
##
## Coefficients:
##           ar1      ar2      ar3  temp_diff_lag1  temp_diff_lag2  dew_diff_lag1
##       -0.3269  -0.2043  -0.1048          0.1020          0.0573        -0.1103
## s.e.   0.0366   0.0379   0.0368          0.0708          0.0710         0.0526
##       dew_diff_lag2  precip_diff_lag1  precip_diff_lag2  fourier(K = 3)C1_52
##             -0.0766            0.0013            0.0015               2.2967
## s.e.         0.0525            0.0006            0.0006               0.2764
##       fourier(K = 3)S1_52  fourier(K = 3)C2_52  fourier(K = 3)S2_52
##                   -1.1927              -0.1284              -0.1038
## s.e.               0.2764               0.1413               0.1407
##       fourier(K = 3)C3_52  fourier(K = 3)S3_52
##                   -0.0283              -0.1428
## s.e.               0.0974               0.0973
##
## sigma^2 estimated as 1.113:  log likelihood=-1093.85
## AIC=2219.69   AICc=2220.44   BIC=2293.62
```

```
## Plot Residuals
dengue_sj_arima2 %>%
  gg_tsresiduals(lag_max = 52) +
  labs(title = "ARIMA w/ Errors Model Residuals",
       subtitle = "San Juan, Puerto Rico")
```



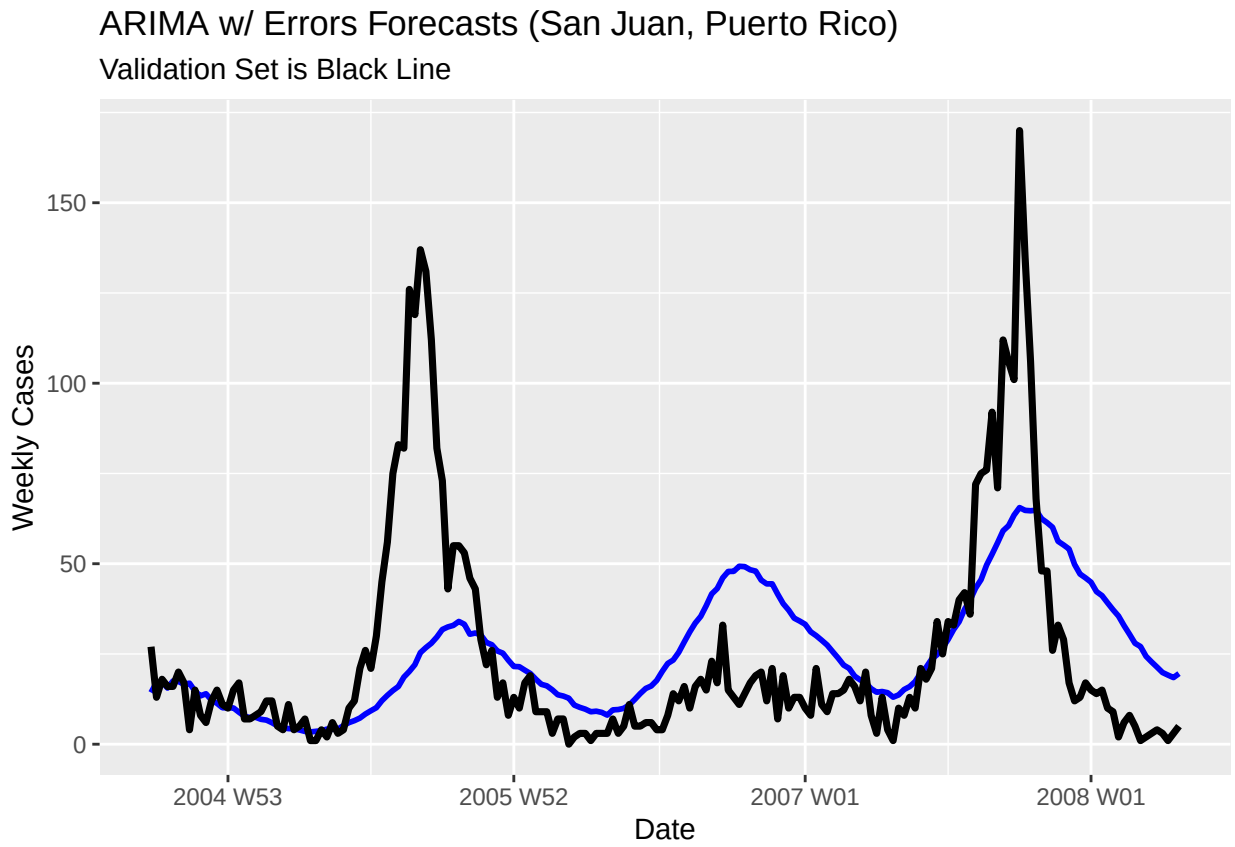
```
## Ljung-Box Test
dengue_sj_arima2 %>%
  augment() %>%
  features(.innov, ljung_box, lag = 52)
```

```
## # A tibble: 1 x 4
##   city  .model          lb_stat lb_pvalue
##   <chr> <chr>             <dbl>     <dbl>
## 1 sj    ARIMA w/ Errors    49.3     0.581
```

```
## Accuracy of Model (focus on MAE since that is what the competition uses)
dengue_sj_arima2_forecast = dengue_sj_arima2 %>%
  forecast(tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8)))

dengue_sj_arima2_forecast %>%
  accuracy(tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8))) %>%
  select(.model, MAE)
```

```
## # A tibble: 1 x 2
##   .model            MAE
##   <chr>           <dbl>
## 1 ARIMA w/ Errors  17.9
```

```
## Plot Forecast
dengue_sj_arima2_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "ARIMA w/ Errors Forecasts (San Juan, Puerto Rico)",
       subtitle = "Validation Set is Black Line") +
  geom_line(data = tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8)),
            aes(x = week, y = total_cases),
            lwd = 1.25)
```



Across both cities the model fits much better than the benchmark with AIC values of 1517 and 2220 for Iquitos and San Juan respectively. The accuracy of the model is slightly worse with the data from Iquito, but significantly improved from San Juan. The MAE is now more consistant across both cities.

Neural Network

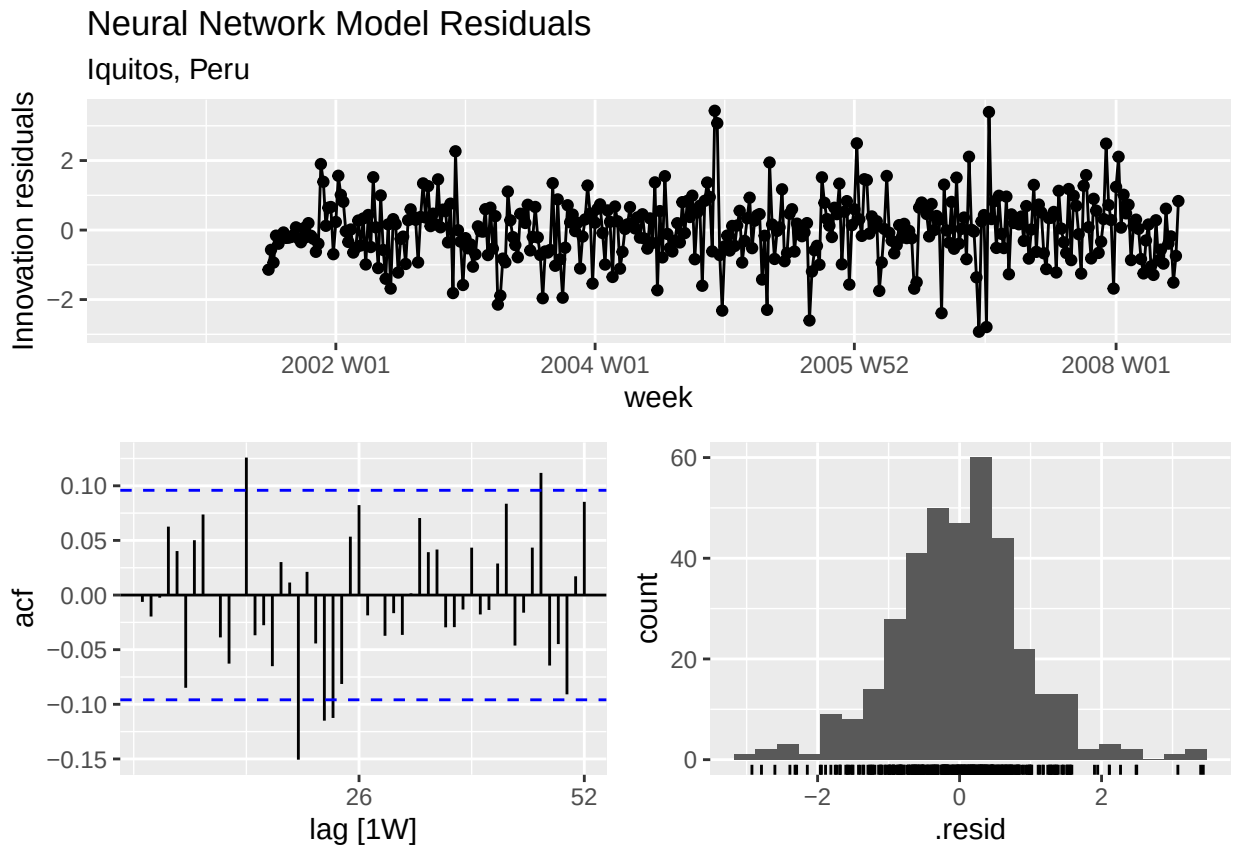
The last model I create is a neural network using the same predictors I used in the second ARIMA model.

```
# Iquitos
## Model
dengue_iq_nn = head(dengue_iq_train, round(train_iq_obs*0.8)) %>%
  model(
    NN = NNETAR(box_cox(total_cases, lambda) ~ temp_diff_lag1 + temp_diff_lag2 +
                  dew_diff_lag1 + dew_diff_lag2 +
                  precip_diff_lag1 + precip_diff_lag2)
  )

## Report
report(dengue_iq_nn)
```

```
## Series: total_cases
## Model: NNAR(3,1,6)[52]
## Transformation: box_cox(total_cases, lambda)
##
## Average of 20 networks, each of which is
```

```
## a 10-6-1 network with 73 weights
## options were - linear output units
##
## sigma^2 estimated as 0.8463
```

```
## Plot Residuals
dengue_iq_nn %>%
  gg_tsresiduals(lag_max = 52) +
  labs(title = "Neural Network Model Residuals",
       subtitle = "Iquitos, Peru")
```




```
## Ljung-Box Test
dengue_iq_nn %>%
  augment() %>%
  features(.innov, ljung_box, lag = 52)
```

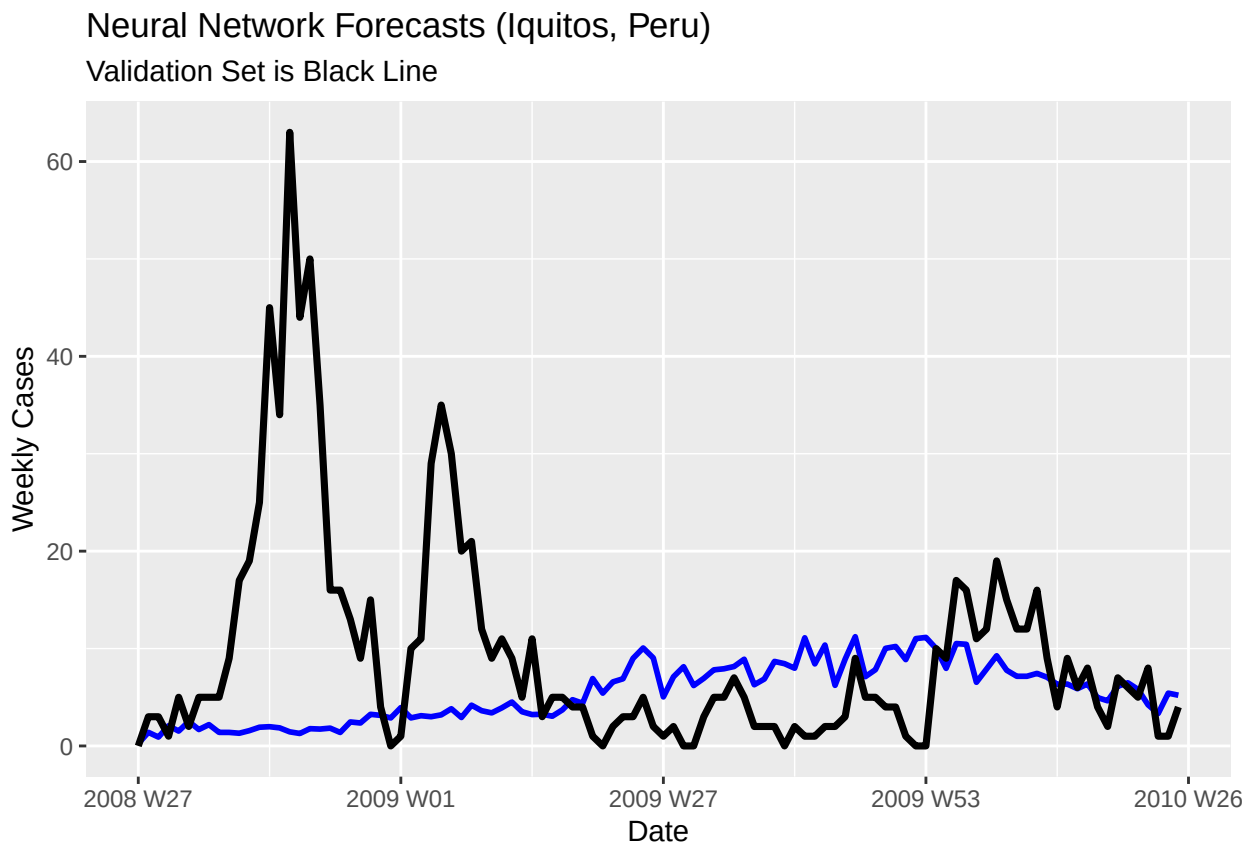
```
## # A tibble: 1 x 4
##   city  .model lb_stat lb_pvalue
##   <chr> <chr>    <dbl>     <dbl>
## 1 iq    NN        71.3    0.0387
```

```
## Accuracy of Model (focus on MAE since that is what the competition uses)
dengue_iq_nn_forecast = dengue_iq_nn %>%
  forecast(tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8)),
```

```
           times = 25, scale = TRUE)

dengue_iq_nn_forecast %>%
  accuracy(tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8))) %>%
  select(.model, MAE)
```

```
## # A tibble: 1 x 2
##   .model   MAE
##   <chr>  <dbl>
## 1 NN      8.22
```

```
## Plot Forecast
dengue_iq_nn_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "Neural Network Forecasts (Iquitos, Peru)",
       subtitle = "Validation Set is Black Line") +
  geom_line(data = tail(dengue_iq_train, train_iq_obs - round(train_iq_obs*0.8)),
            aes(x = week, y = total_cases),
            lwd = 1.25)
```

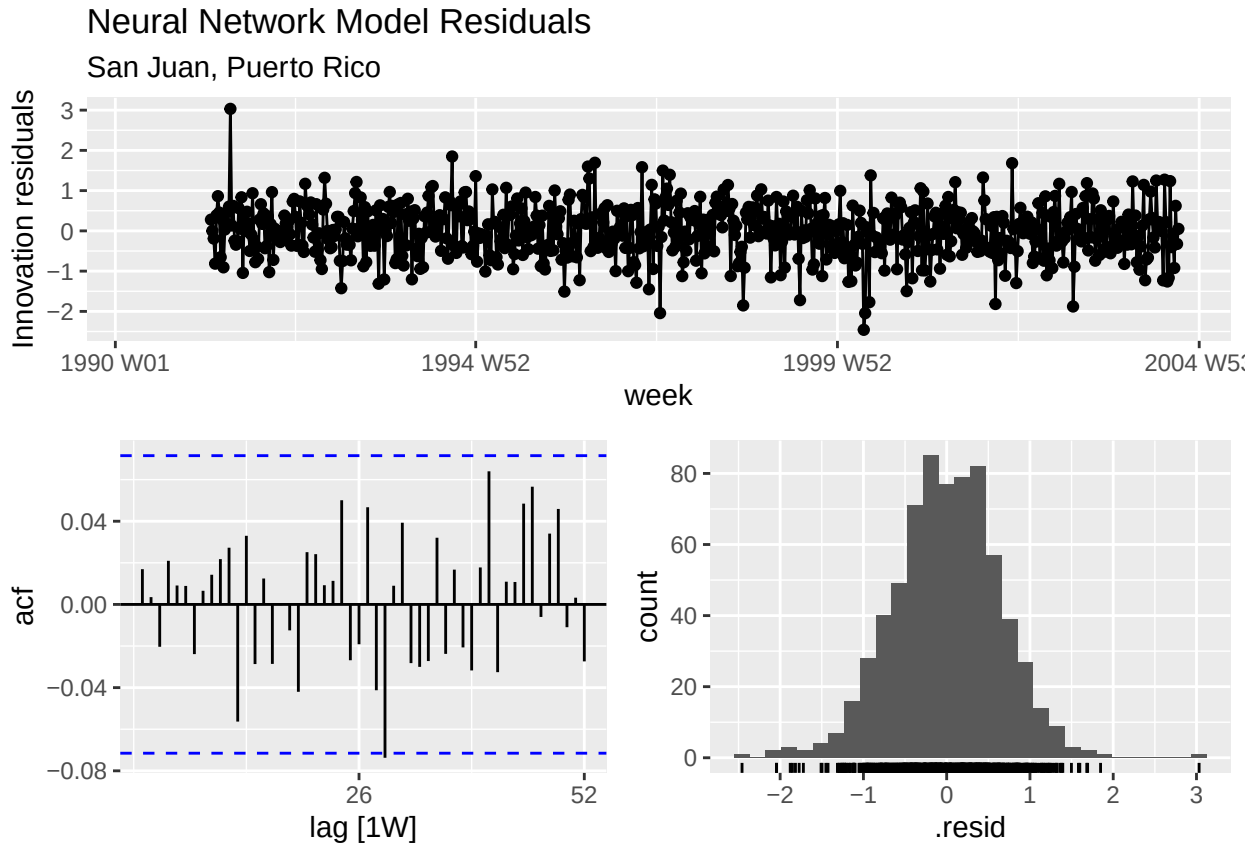



```
# San Juan
## Model
dengue_sj_nn = head(dengue_sj_train, round(train_sj_obs*0.8)) %>%
```

```
  model(
    NN = NNETAR(box_cox(total_cases, lambda) ~ temp_diff_lag1 + temp_diff_lag2 +
                  dew_diff_lag1 + dew_diff_lag2 +
                  precip_diff_lag1 + precip_diff_lag2)
  )

## Report
report(dengue_sj_nn)
```

```
## Series: total_cases
## Model: NNAR(11,1,10)[52]
## Transformation: box_cox(total_cases, lambda)
##
## Average of 20 networks, each of which is
## a 18-10-1 network with 201 weights
## options were - linear output units
##
## sigma^2 estimated as 0.4132
```

```
## Plot Residuals
dengue_sj_nn %>%
  gg_tsresiduals(lag_max = 52) +
  labs(title = "Neural Network Model Residuals",
       subtitle = "San Juan, Puerto Rico")
```



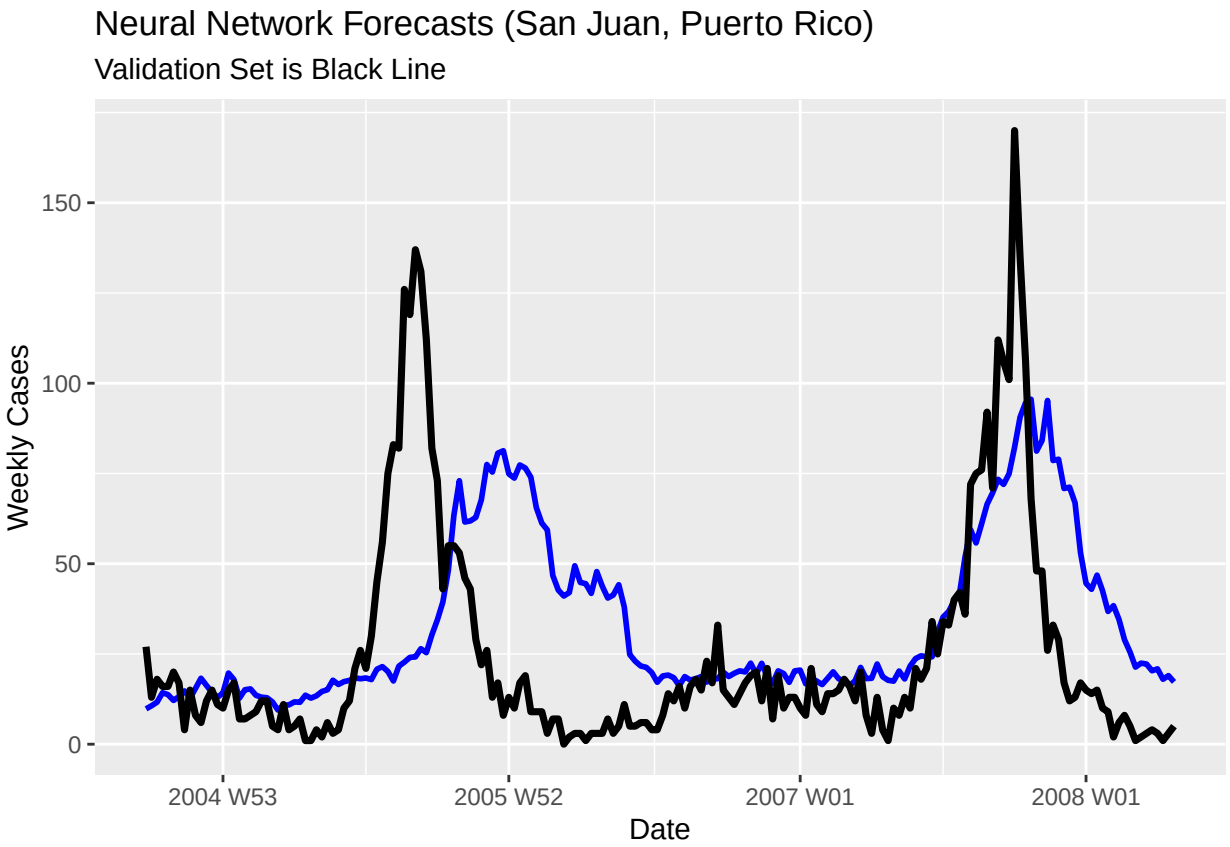
```r
## Ljung-Box Test
dengue_sj_nn %>%
  augment() %>%
  features(.innov, ljung_box, lag = 52)
```

```
## # A tibble: 1 x 4
##   city  .model lb_stat lb_pvalue
##   <chr> <chr>    <dbl>     <dbl>
## 1 sj    NN        35.5     0.961
```

```r
## Accuracy of Model (focus on MAE since that is what the competition uses)
dengue_sj_nn_forecast = dengue_sj_nn %>%
  forecast(tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8)),
           times = 10, scale = TRUE)

dengue_sj_nn_forecast %>%
  accuracy(tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8))) %>%
  select(.model, MAE)
```

```
## # A tibble: 1 x 2
##   .model   MAE
##   <chr>  <dbl>
## 1 NN      22.1
```

```r
## Plot Forecast
dengue_sj_nn_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "Neural Network Forecasts (San Juan, Puerto Rico)",
       subtitle = "Validation Set is Black Line") +
  geom_line(data = tail(dengue_sj_train, train_sj_obs - round(train_sj_obs*0.8)),
            aes(x = week, y = total_cases),
            lwd = 1.25)
```



As with the ARIMA model the predictions are more consistent across cities and overall the MAE is lower than the benchmark. The fit appears better with a lower σ^2 than the ARIMA model. With the models complete I move towards testing the data on the DrivenData website.

Prediction

I prepare the test set the same way I prepared the training set. Split by city. All of the same variable creation or lagging done on the training data is done again on the test data.

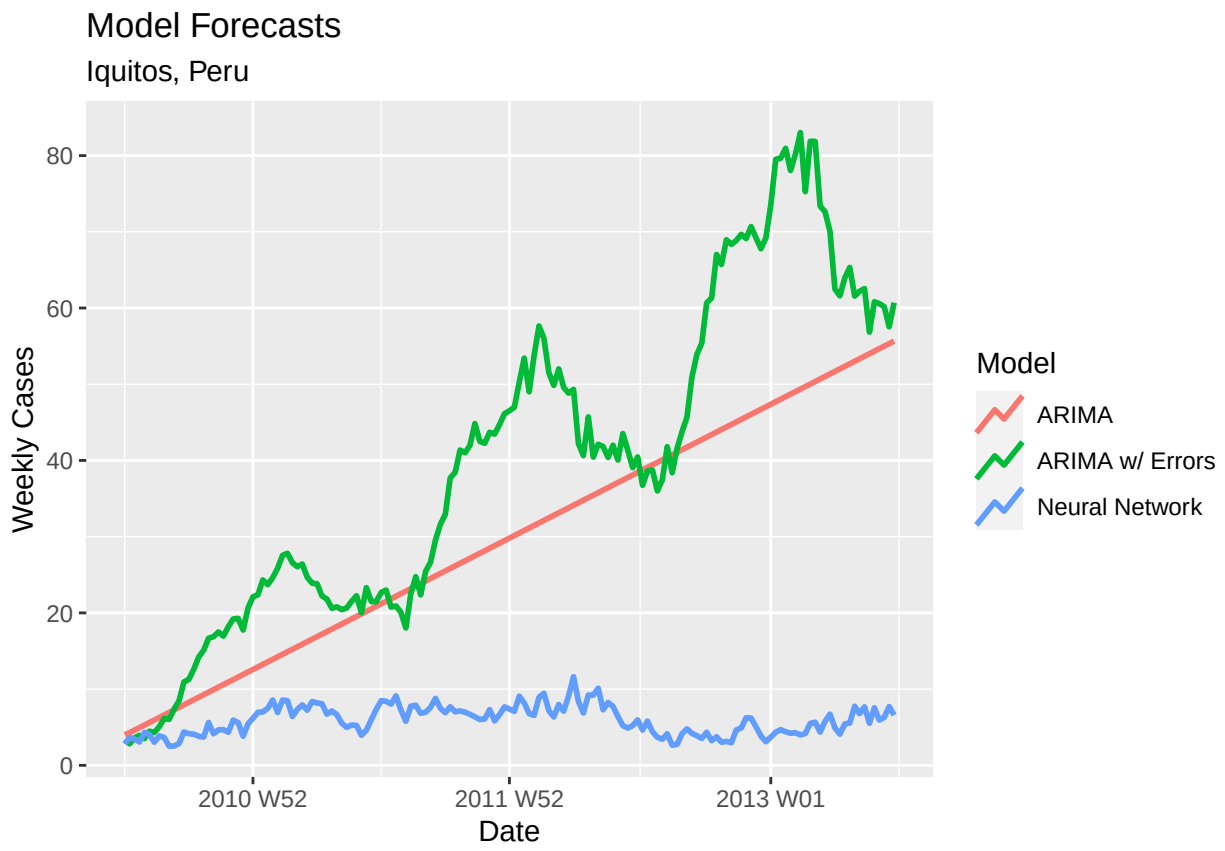
```
# Create test set tsibble
dengue_test = left_join(dengue_features_test, dengue_labels_test,
                        by = c("city", "year", "weekofyear"))

dengue_test = dengue_test %>%
  mutate(week = yearweek(base::as.Date(week_start_date))) %>%
  as_tsibble(index = week, key = city)

# Split into cities
dengue_iq_test = dengue_test %>%
  filter(city == "iq")

dengue_sj_test = dengue_test %>%
  filter(city == "sj")

# Create the differenced values
```

```
dengue_iq_test = bind_rows(extra_iq_train, dengue_iq_test)

dengue_iq_test = dengue_iq_test %>%
  mutate(temp_diff = difference(reanalysis_avg_temp_k),
         dew_diff = difference(reanalysis_dew_point_temp_k),
         precip_diff = difference(reanalysis_sat_precip_amt_mm)) %>%
  mutate(temp_diff_lag1 = lag(temp_diff),
         temp_diff_lag2 = lag(temp_diff, 2),
         dew_diff_lag1 = lag(dew_diff),
         dew_diff_lag2 = lag(dew_diff, 2),
         precip_diff_lag1 = lag(precip_diff),
         precip_diff_lag2 = lag(precip_diff, 2))

dengue_sj_test = bind_rows(extra_sj_train, dengue_sj_test)

dengue_sj_test = dengue_sj_test %>%
  mutate(temp_diff = difference(reanalysis_avg_temp_k),
         dew_diff = difference(reanalysis_dew_point_temp_k),
         precip_diff = difference(reanalysis_sat_precip_amt_mm)) %>%
  mutate(temp_diff_lag1 = lag(temp_diff),
         temp_diff_lag2 = lag(temp_diff, 2),
         dew_diff_lag1 = lag(dew_diff),
         dew_diff_lag2 = lag(dew_diff, 2),
         precip_diff_lag1 = lag(precip_diff),
         precip_diff_lag2 = lag(precip_diff, 2))

# Fill gaps
dengue_iq_test = dengue_iq_test %>%
  fill_gaps() %>%
  mutate_all( ~ na.locf(.x, na.rm = FALSE))

dengue_sj_test = dengue_sj_test %>%
  fill_gaps() %>%
  mutate_all( ~ na.locf(.x, na.rm = FALSE))

# Remove training observations
dengue_iq_test = tail(dengue_iq_test, -3)
dengue_sj_test = tail(dengue_sj_test, -3)
```

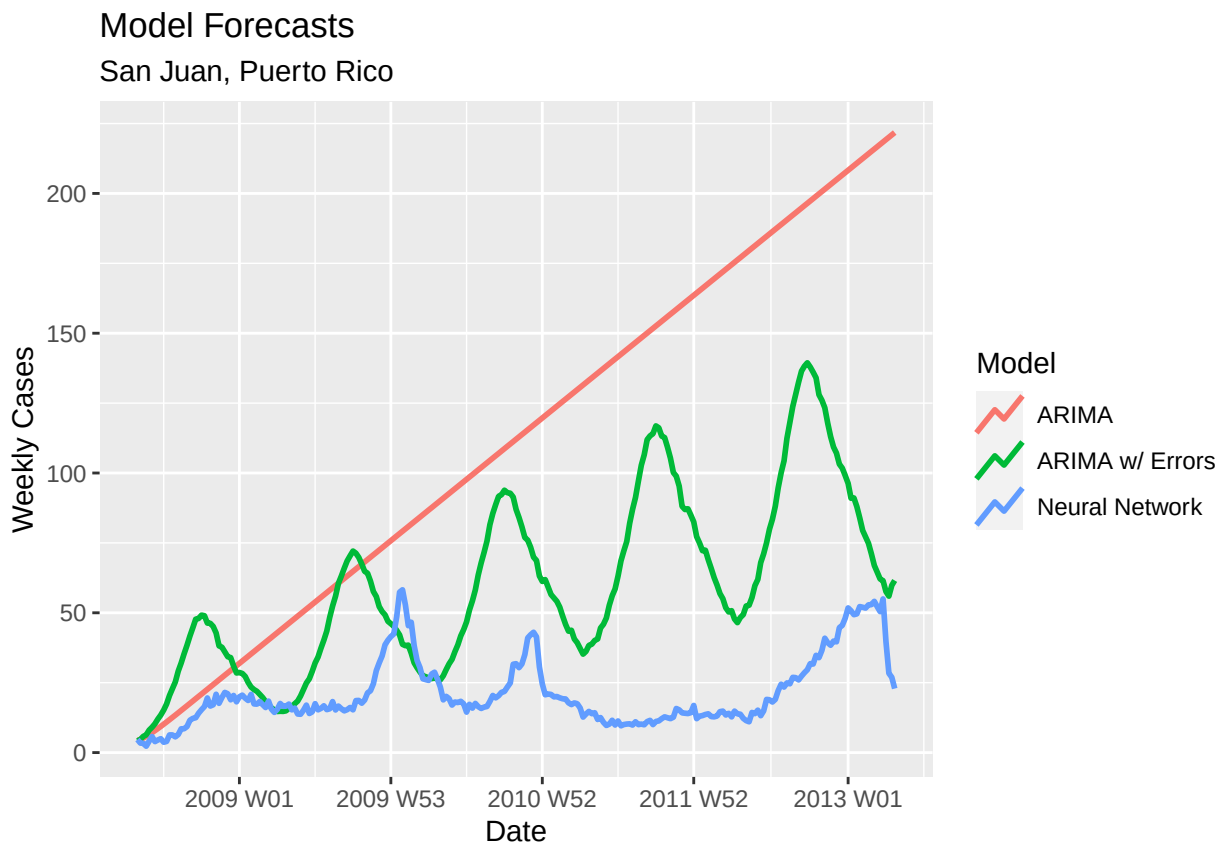
I add the last three observations from each city's training data to create the differenced and lagged values. There were issues forecasting with the neural network because of missing predictor values for any part of the time series.

```
# Iquitos
dengue_iq_model = dengue_iq_train %>%
  model(
    ARIMA = ARIMA(box_cox(total_cases, lambda) ~ pdq(1,1,3) + PDQ(0,0,0)),
    `ARIMA w/ Errors` = ARIMA(box_cox(total_cases, lambda) ~ temp_diff_lag1 +
                                temp_diff_lag2 + dew_diff_lag1 + dew_diff_lag2 +
                                precip_diff_lag1 + precip_diff_lag2 +
                                pdq(3,1,0) + PDQ(0,0,0) + fourier(K = 3)),
    `Neural Network` = NNETAR(box_cox(total_cases, lambda) ~ temp_diff_lag1 +
                                temp_diff_lag2 + dew_diff_lag1 + dew_diff_lag2 +
```

```
                                precip_diff_lag1 + precip_diff_lag2)
  )

# San Juan
dengue_sj_model = dengue_sj_train %>%
  model(
    ARIMA = ARIMA(box_cox(total_cases, lambda) ~ pdq(1,1,3) + PDQ(0,0,0)),
    `ARIMA w/ Errors` = ARIMA(box_cox(total_cases, lambda) ~ temp_diff_lag1 +
                                temp_diff_lag2 + dew_diff_lag1 + dew_diff_lag2 +
                                precip_diff_lag1 + precip_diff_lag2 +
                                pdq(3,1,0) + PDQ(0,0,0) + fourier(K = 3)),
    `Neural Network` = NNETAR(box_cox(total_cases, lambda) ~ temp_diff_lag1 +
                                temp_diff_lag2 + dew_diff_lag1 + dew_diff_lag2 +
                                precip_diff_lag1 + precip_diff_lag2)
  )

# Predict
dengue_iq_forecast = dengue_iq_model %>%
  forecast(dengue_iq_test, times = 25, scale = TRUE)

dengue_sj_forecast = dengue_sj_model %>%
  forecast(dengue_sj_test, times = 25, scale = TRUE)

# Plot
dengue_iq_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "Model Forecasts",
       subtitle = "Iquitos, Peru") +
  guides(color = guide_legend(title = "Model"))
```




```
dengue_sj_forecast %>%
  autoplot(level = NULL, lwd = 1) +
  labs(x = "Date", y = "Weekly Cases",
       title = "Model Forecasts",
       subtitle = "San Juan, Puerto Rico") +
  guides(color = guide_legend(title = "Model"))
```



With the predictions in hand. I create the csv files for submission. Each model was submitted to the DrivenData website and scored.

```
# Split label data into each respective city
dengue_iq_format = dengue_labels_test %>%
  filter(city == "iq")

dengue_sj_format = dengue_labels_test %>%
  filter(city == "sj")

# Split forecasts into each respective model and join with the format object
## Iquitos
arima_iq = dengue_iq_forecast %>%
  filter(.model == "ARIMA") %>%
  distinct(week_start_date, .keep_all = TRUE)

arima_iq = data.frame(
  weekofyear = arima_iq$weekofyear,
  year = arima_iq$year,
  city = arima_iq$city,
  arima = round(arima_iq$.mean)
)

arima2_iq = dengue_iq_forecast %>%
  filter(.model == "ARIMA w/ Errors") %>%
  distinct(week_start_date, .keep_all = TRUE)
```

```r
arima2_iq = data.frame(
  weekofyear = arima2_iq$weekofyear,
  year = arima2_iq$year,
  city = arima2_iq$city,
  arima2 = round(arima2_iq$.mean)
)

nn_iq = dengue_iq_forecast %>%
  filter(.model == "Neural Network") %>%
  distinct(week_start_date, .keep_all = TRUE)

nn_iq = data.frame(
  weekofyear = nn_iq$weekofyear,
  year = nn_iq$year,
  city = nn_iq$city,
  nn = round(nn_iq$.mean)
)

dengue_iq_format = left_join(dengue_iq_format, arima_iq,
                             by = c("city", "year", "weekofyear"))
dengue_iq_format = left_join(dengue_iq_format, arima2_iq,
                             by = c("city", "year", "weekofyear"))
dengue_iq_format = left_join(dengue_iq_format, nn_iq,
                             by = c("city", "year", "weekofyear"))

## San Juan
arima_sj = dengue_sj_forecast %>%
  filter(.model == "ARIMA") %>%
  distinct(week_start_date, .keep_all = TRUE)

arima_sj = data.frame(
  weekofyear = arima_sj$weekofyear,
  year = arima_sj$year,
  city = arima_sj$city,
  arima = round(arima_sj$.mean)
)

arima2_sj = dengue_sj_forecast %>%
  filter(.model == "ARIMA w/ Errors") %>%
  distinct(week_start_date, .keep_all = TRUE)

arima2_sj = data.frame(
  weekofyear = arima2_sj$weekofyear,
  year = arima2_sj$year,
  city = arima2_sj$city,
  arima2 = round(arima2_sj$.mean)
)

nn_sj = dengue_sj_forecast %>%
  filter(.model == "Neural Network") %>%
  distinct(week_start_date, .keep_all = TRUE)

nn_sj = data.frame(
```

```
  weekofyear = nn_sj$weekofyear,
  year = nn_sj$year,
  city = nn_sj$city,
  nn = round(nn_sj$.mean)
)

dengue_sj_format = left_join(dengue_sj_format, arima_sj,
                             by = c("city", "year", "weekofyear"))
dengue_sj_format = left_join(dengue_sj_format, arima2_sj,
                             by = c("city", "year", "weekofyear"))
dengue_sj_format = left_join(dengue_sj_format, nn_sj,
                             by = c("city", "year", "weekofyear"))

# Combine into final product and write to disk
submission = rbind(dengue_sj_format, dengue_iq_format)
write.csv(submission, file.path(path, "submission.csv"), row.names = FALSE)
```

Limitations

The biggest limitation of the models using predictors is the forecasting horizon. Since I lagged predictors by only one week, I would reasonably only be able to forecast cases out a week unless I start forecasting predictors as well. The model can be evaluated on the test set, but practical application would be limited.

Conclusions

DrivenData scored all three models. My benchmark model had an MAE of 54.04. This is likely due to the San Juan predictions as the model had an MAE of more than 90 on the validation set. The ARIMA w/ Errors model fared better with a MAE of 37.53. The Neural Network performed the best with an MAE of 28.2. All submissions were done under the username "ekenney" and the submission IDs are id-240761, id-240832, and id-240833. A screenshot is included with this paper for submission. For comparison, the top performing model on this test set had an MAE of 10.1. Future work I would recommend is continuing with the Neural Network to see what adjustments could be made to lower the MAE even more. Additionally, I would be interested to see features and case data from another city to see how the models perform in a general sense.

References

Abualamah, W. A., Akbar, N. A., Banni, H. S., & Bafail, M. A. (2021). Forecasting the morbidity and mortality of dengue fever in KSA: A time series analysis (2006–2016). *Journal of Taibah University Medical Sciences*, 16(3), 448–455. https://doi.org/10.1016/j.jtumed.2021.02.007

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice*. OTexts.

Kularatne, S. A. (2015). Dengue fever. *BMJ*. https://doi.org/10.1136/bmj.h4661