Predictive Analytics & Forecasting - Final

Eric Kenney

2023-08-12

Abstract

Problem Statement

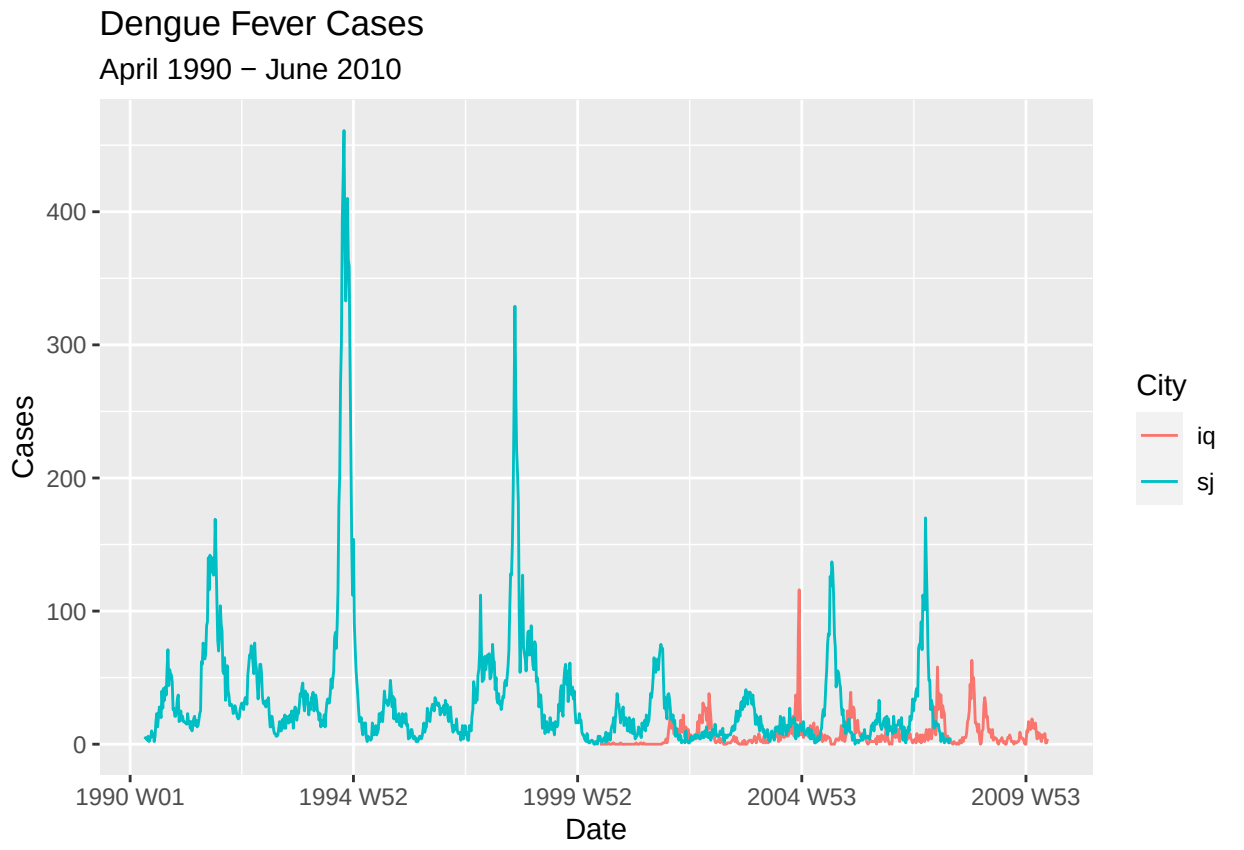
Using data from the DengAI: Predicting Disease Spread project from DrivenData I model and predict weekly cases of dengue fever in the cities of San Juan, Puerto Rico and Iquitos, Peru. Dengue fever is a tropical disease spread by mosquitos. In most cases, it resembles the flu with symptoms that include 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.

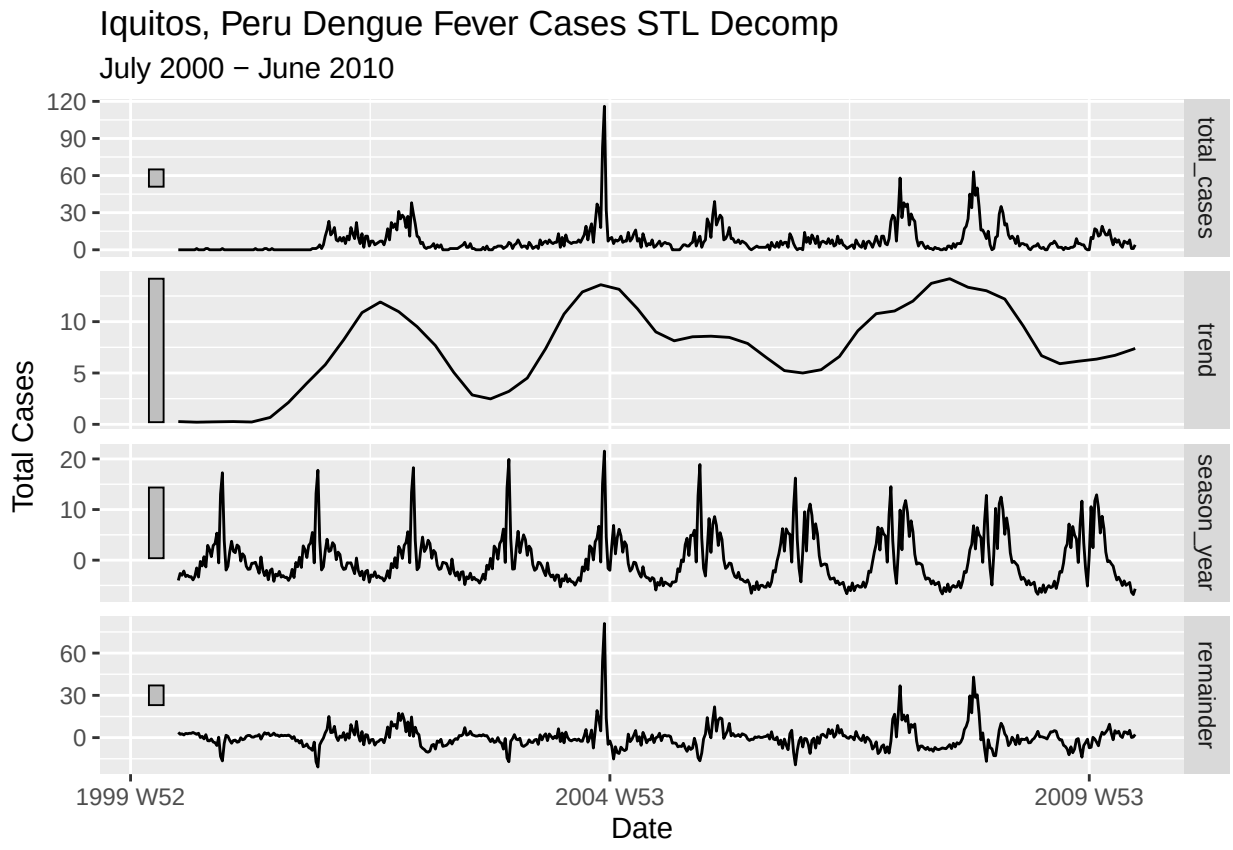
```
# Read in data
dengue_features_train = read.csv(file.path(path, "dengue_features_train.csv"))
dengue_features_test = read.csv(file.path(path, "dengue_features_test.csv"))
dengue_labels_train = read.csv(file.path(path, "dengue_labels_train.csv"))

# 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"))
```

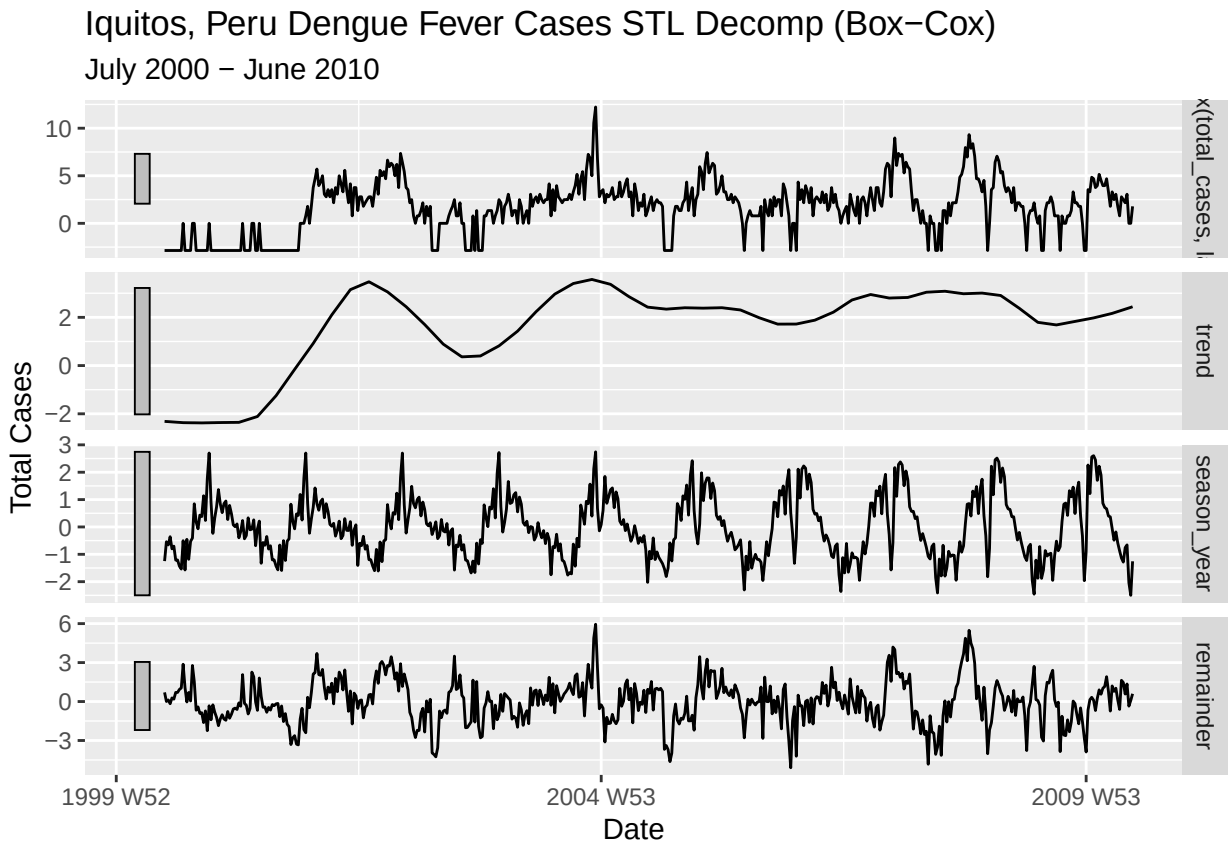



```
# Create training sets based on city
dengue_iq_train = dengue_train %>%
  filter(city == "iq")

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))
```

```
# 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")
```

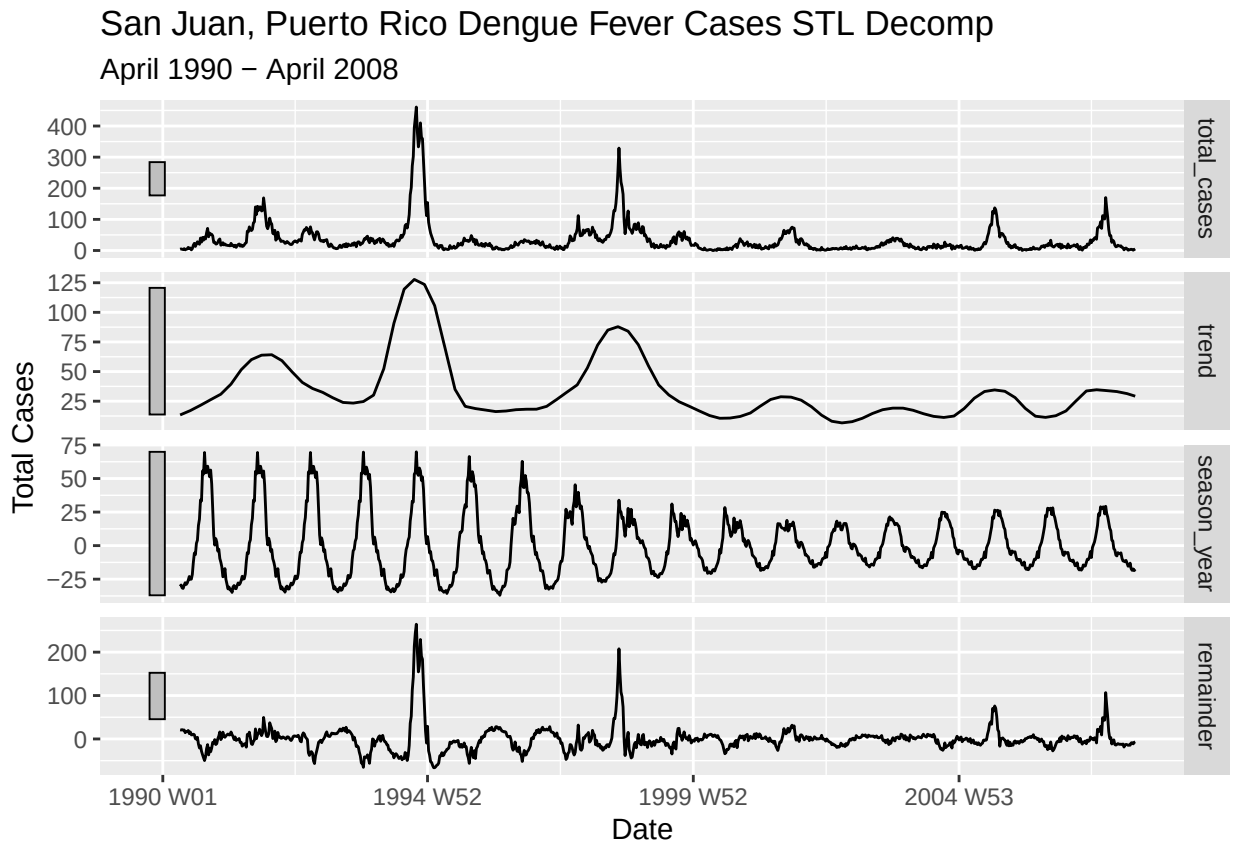



```
## Box-Cox
dengue_iq_train %>%
  model(STL(box_cox(total_cases, lambda))) %>%
  components() %>%
  autoplot() +
  labs(x = "Date", y = "Total Cases",
       title = "Iquitos, Peru Dengue Fever Cases STL Decomp (Box-Cox)",
       subtitle = "July 2000 - June 2010")
```

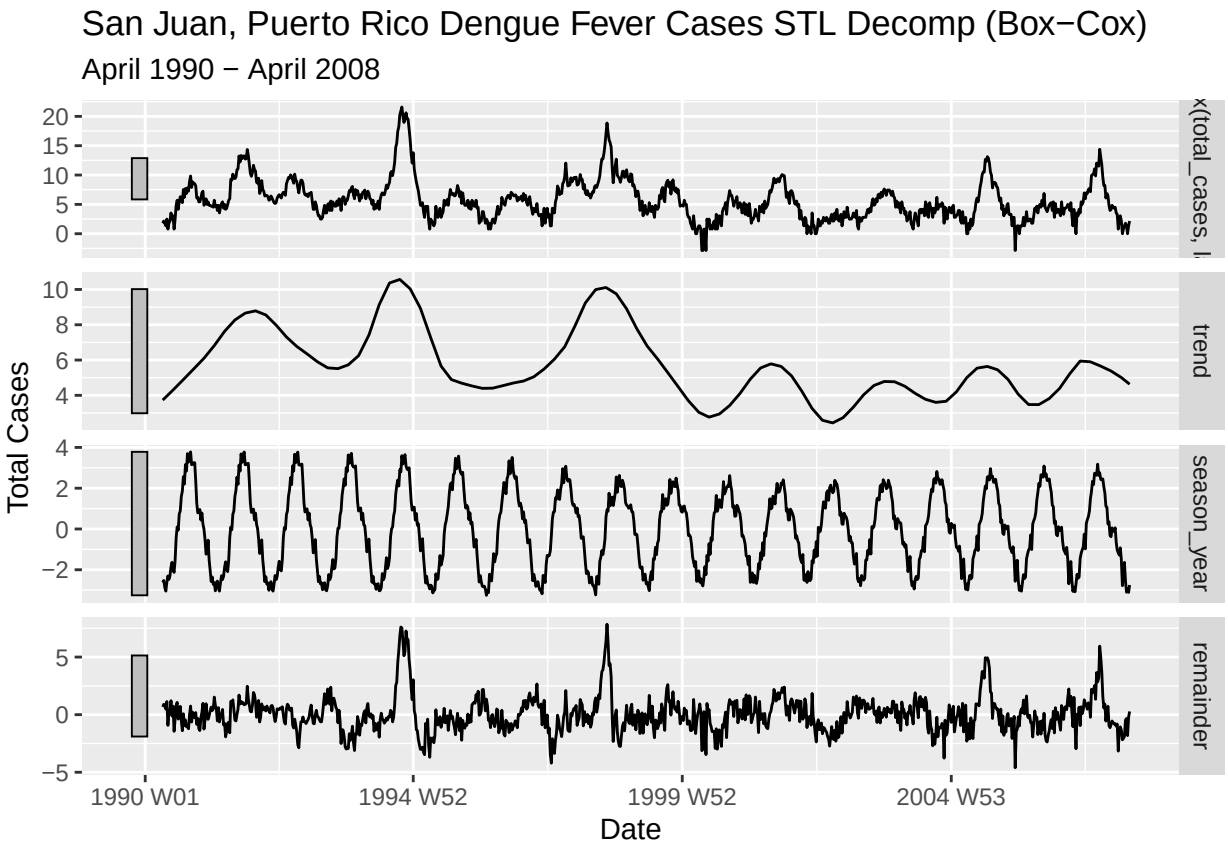


The data from Iquitos had consistent trend and seasonal effects, but due to seasonal variance in the data from San Juan I performed a Box-Cox transformation on the Iquitos data. As seen in the above plots, the data is still stable.

```
# 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 transformed 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.

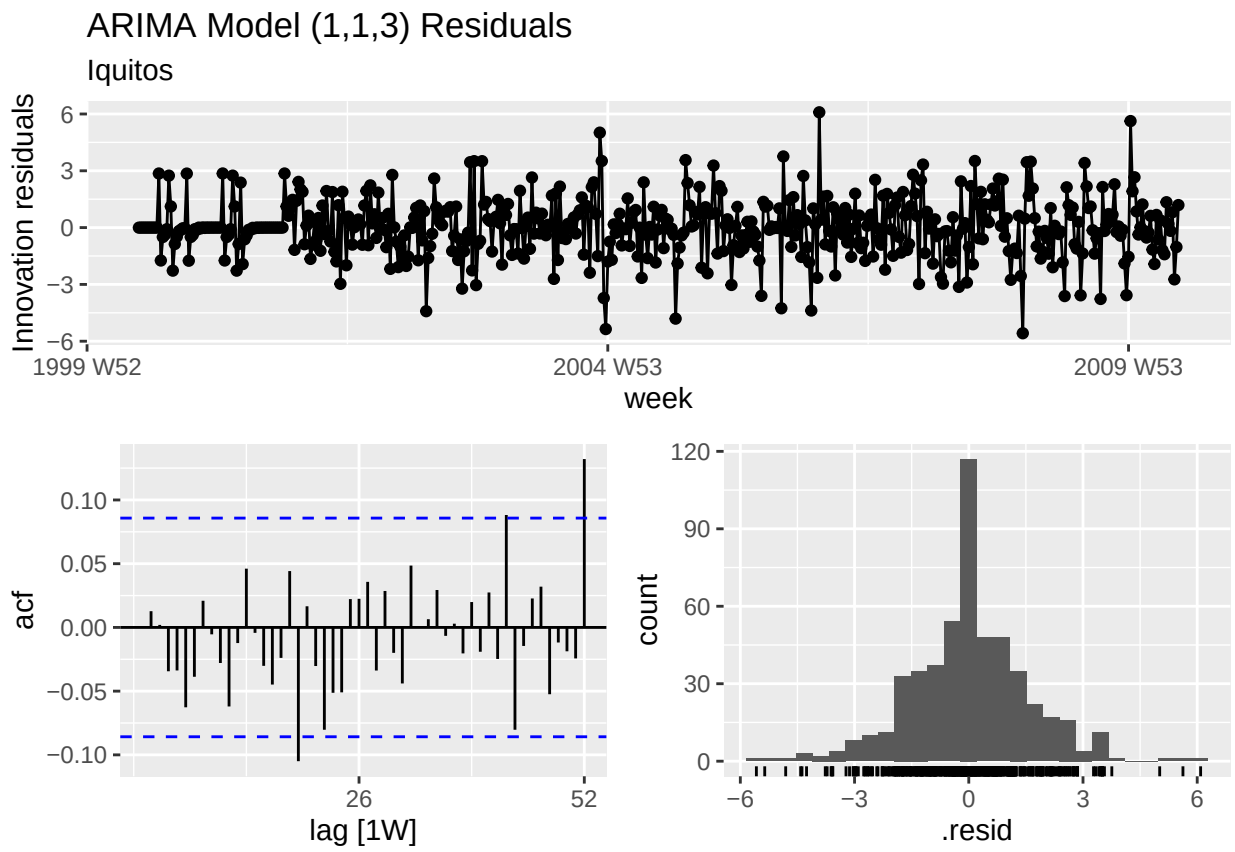
ARIMA (Benchmark Model)

```
# Iquitos
## Create Model
dengue_iq_arima = dengue_iq_train %>%
  model(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
##       -0.7038  0.3120  -0.3403  -0.0199
## s.e.   0.4002  0.4008   0.1593   0.0583
##
## sigma^2 estimated as 2.388:  log likelihood=-964.06
## AIC=1938.12   AICc=1938.24   BIC=1959.4
```

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



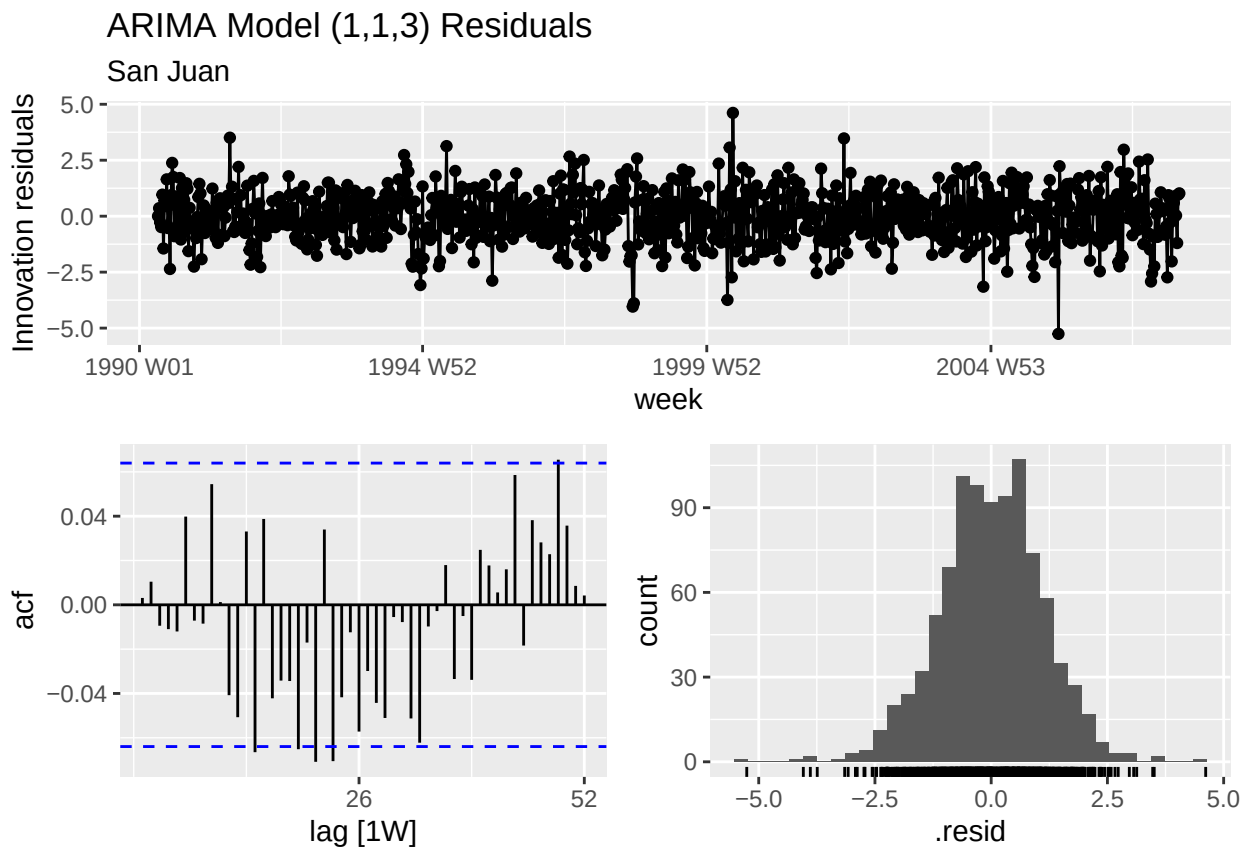

```
## Ljung-Box test
augment(dengue_iq_arima) %>%
  features(.innov, ljung_box, lag = 52)
```

```
## # A tibble: 1 x 4
##   city  .model                                                  lb_stat lb_pvalue
##   <chr> <chr>                                                     <dbl>     <dbl>
## 1 iq    "ARIMA(box_cox(total_cases, lambda) ~ pdq(1, 1, 3) + ~    52.6     0.452
```

```
# San Juan
## Create Model
dengue_sj_arima = dengue_sj_train %>%
  model(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
##       0.7864  -1.0805  0.1785  0.1421
## s.e.  0.0541   0.0608  0.0509  0.0340
##
## sigma^2 estimated as 1.258:  log likelihood=-1436.75
## AIC=2883.5   AICc=2883.56   BIC=2907.71
```

```
## 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(box_cox(total_cases, lambda) ~ pdq(1, 1, 3) + ~    67.3    0.0748
```

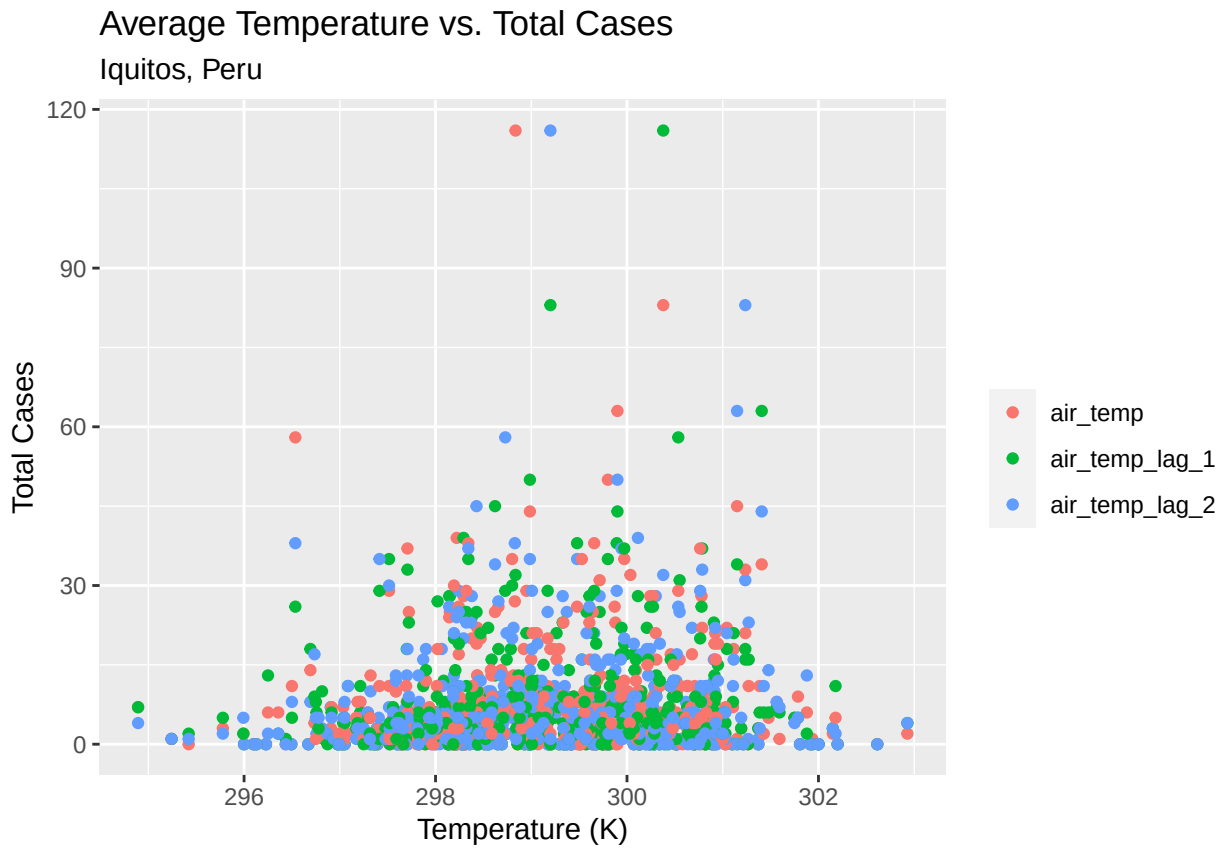
I created two non-seasonal ARIMA models choosing values of $p = 1, d = 1, q = 3$. The model fits the Iquitos data better than San Juan (AIC of 1938 and 2884 respectively). Plotting residuals from both sets of training data show potential issues with autocorrelation, but Ljung-Box tests on both sets show no significant problems.

ARIMA with Predictors

Next, I continue with the ARIMA model and attempted to find a better fit and potential forecast by adding predictors based on previous work regarding 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)

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

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

```
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
```

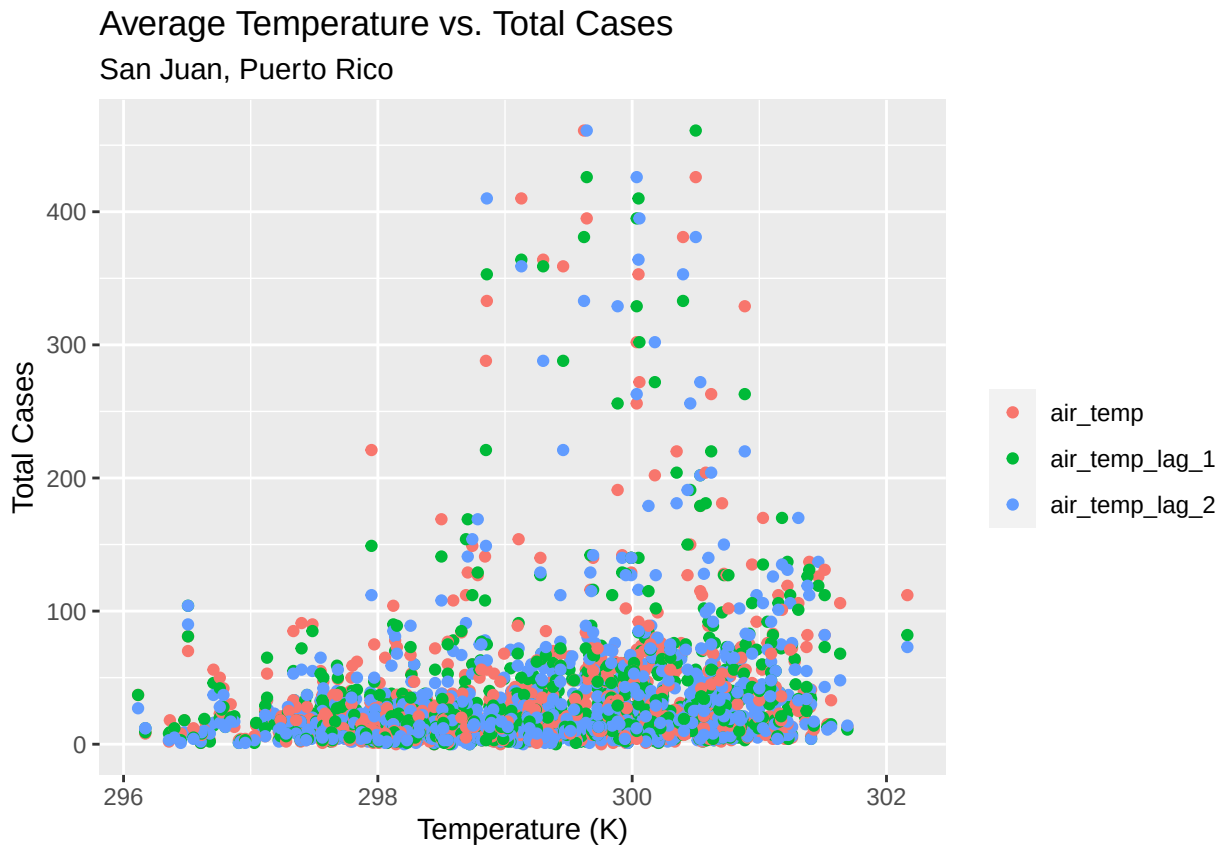
```r
## Stationarity Test
dengue_iq_train %>%
  features(reanalysis_avg_temp_k, unitroot_kpss)
```

```
## # 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)
```

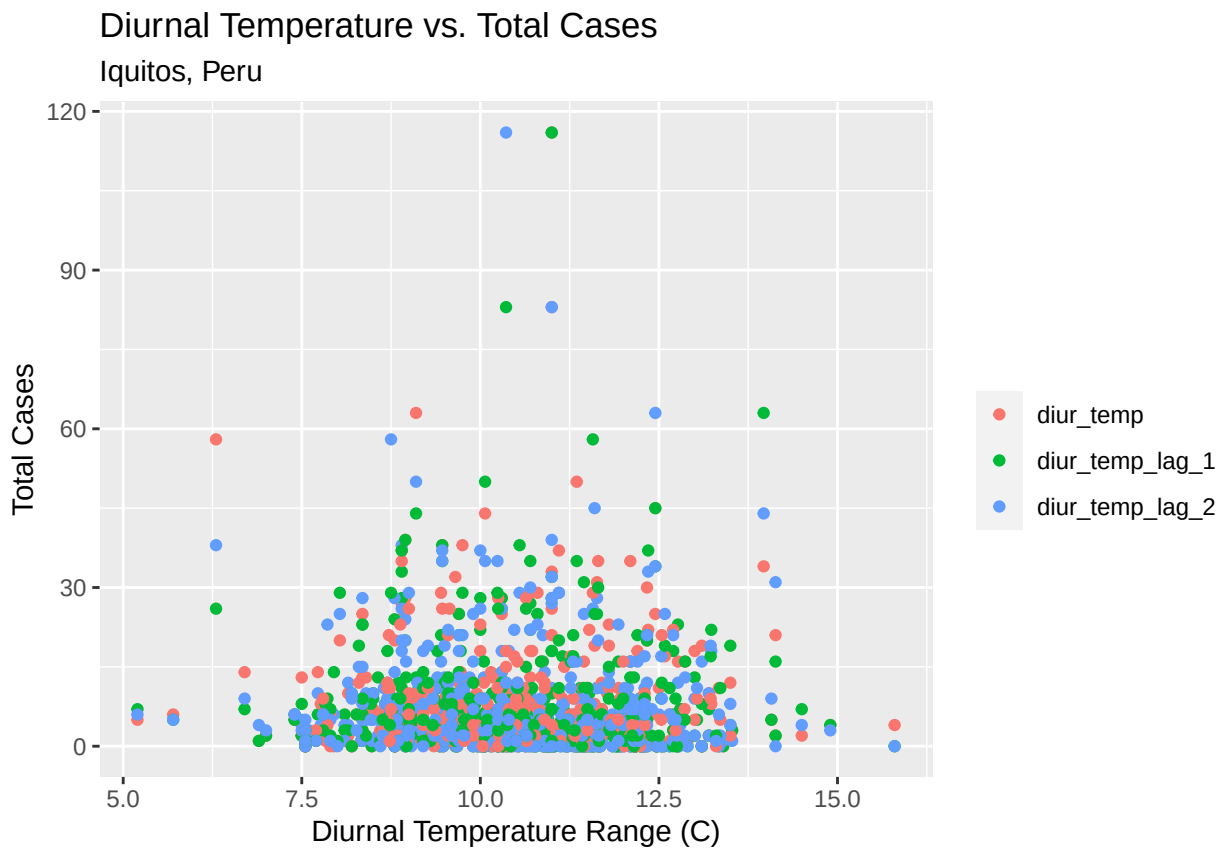
```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj        0.822        0.01
```

```
### 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
```

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 differenced 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
```

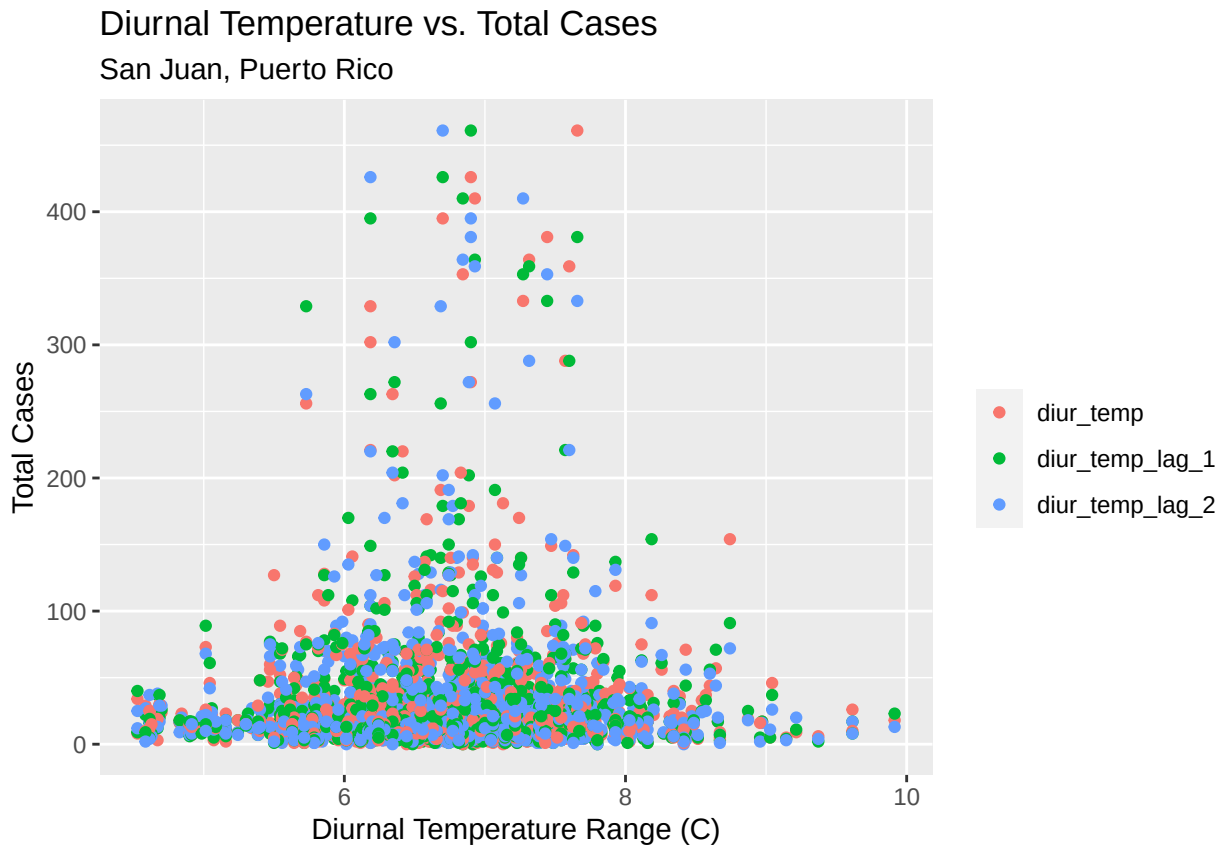
```r
## Stationarity
dengue_sj_train %>%
  features(station_diur_temp_rng_c, unitroot_kpss)
```

```
## # 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)
```

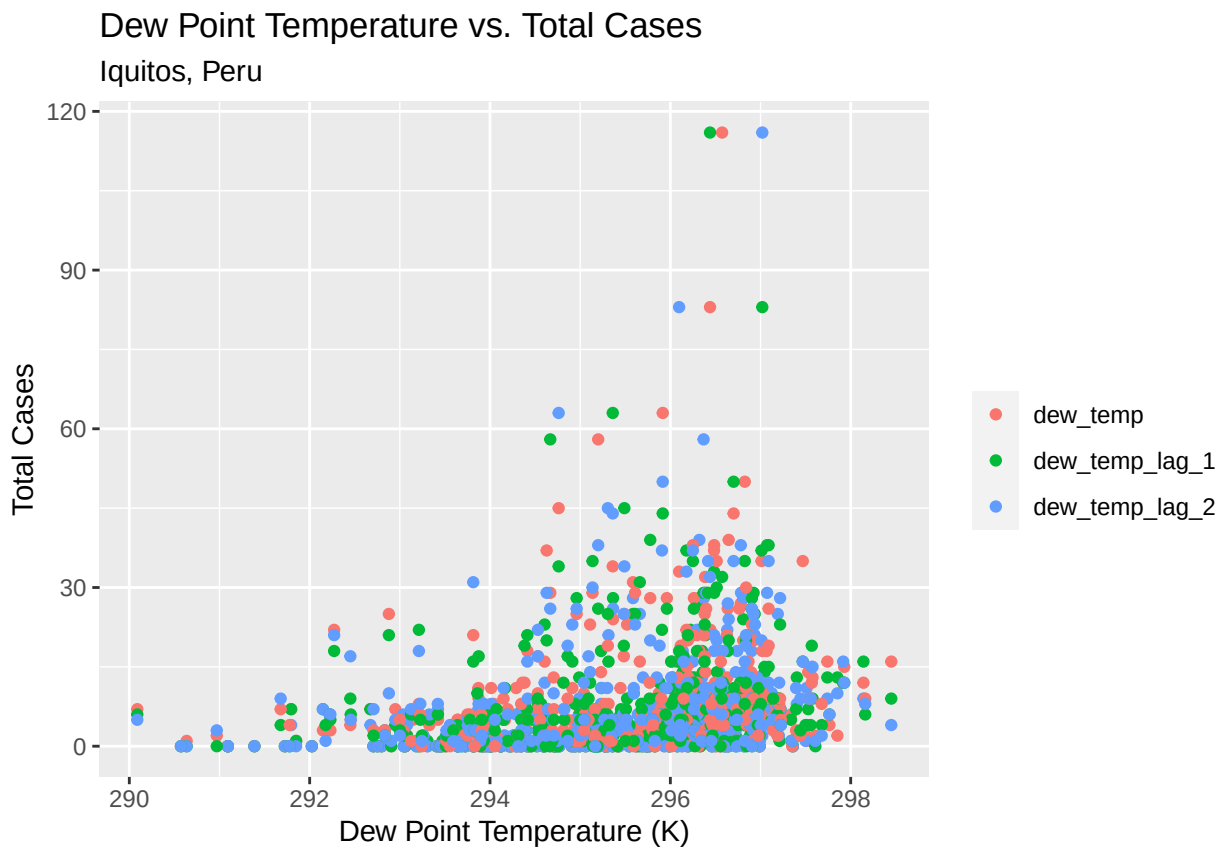
```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj         2.37        0.01
```

```
### 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
```

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
```

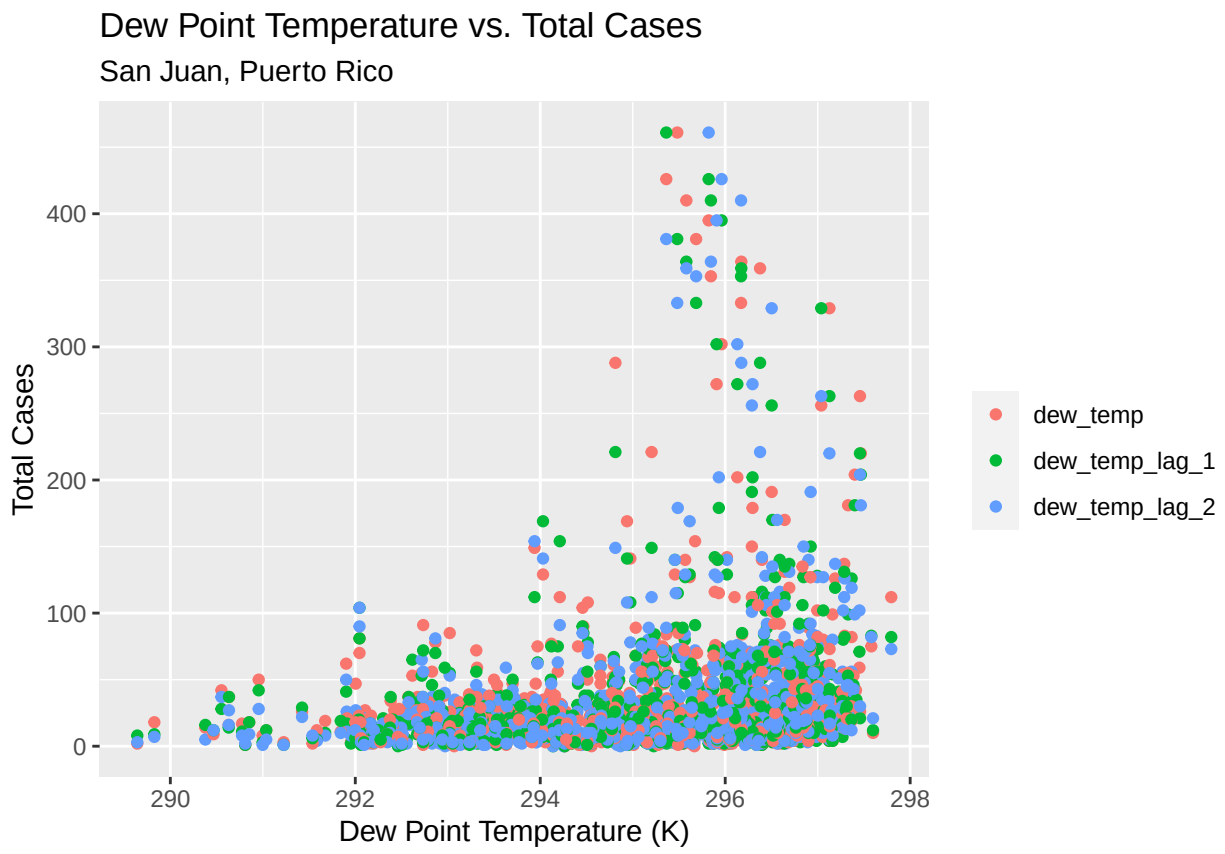
```r
## Stationarity test
dengue_iq_train %>%
  features(reanalysis_dew_point_temp_k, unitroot_kpss)
```

```
## # 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)
```

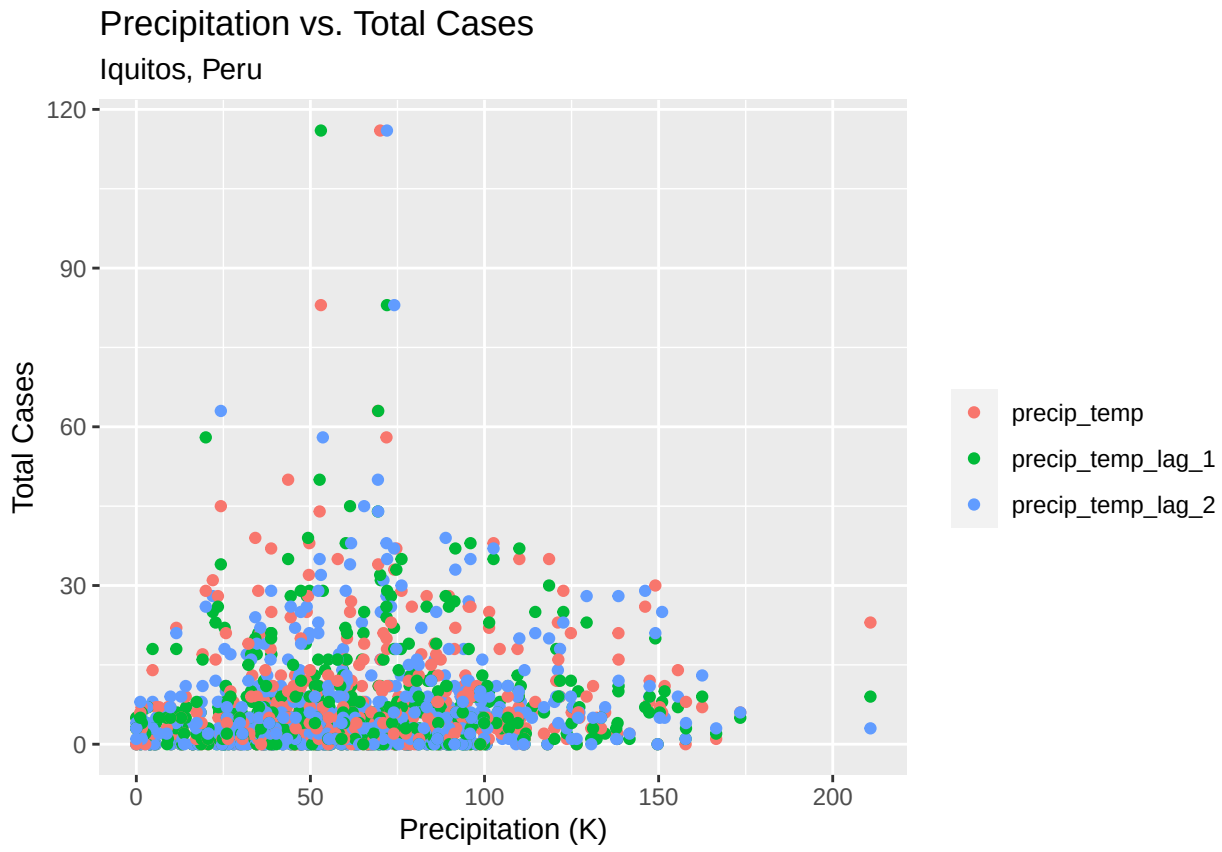
```
## # A tibble: 1 x 3
##   city  kpss_stat kpss_pvalue
##   <chr>     <dbl>       <dbl>
## 1 sj       0.0802         0.1
```

```
### 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
```

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
```

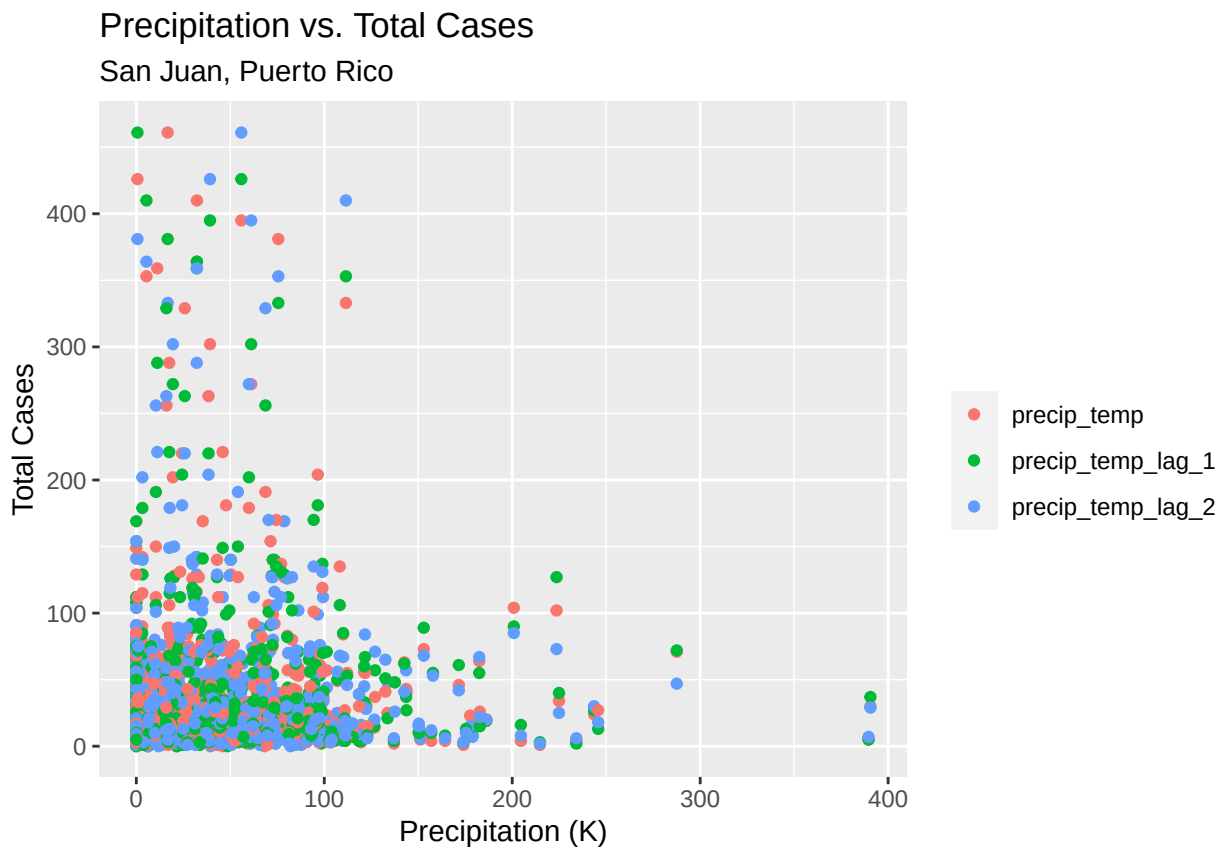
```
print(matrix(data = precip_iq_cor, nrow = 1, ncol = 3,
             dimnames = list("Correlation", c("Base", "Lag 1", "Lag 2"))))
```

```
##                   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
```

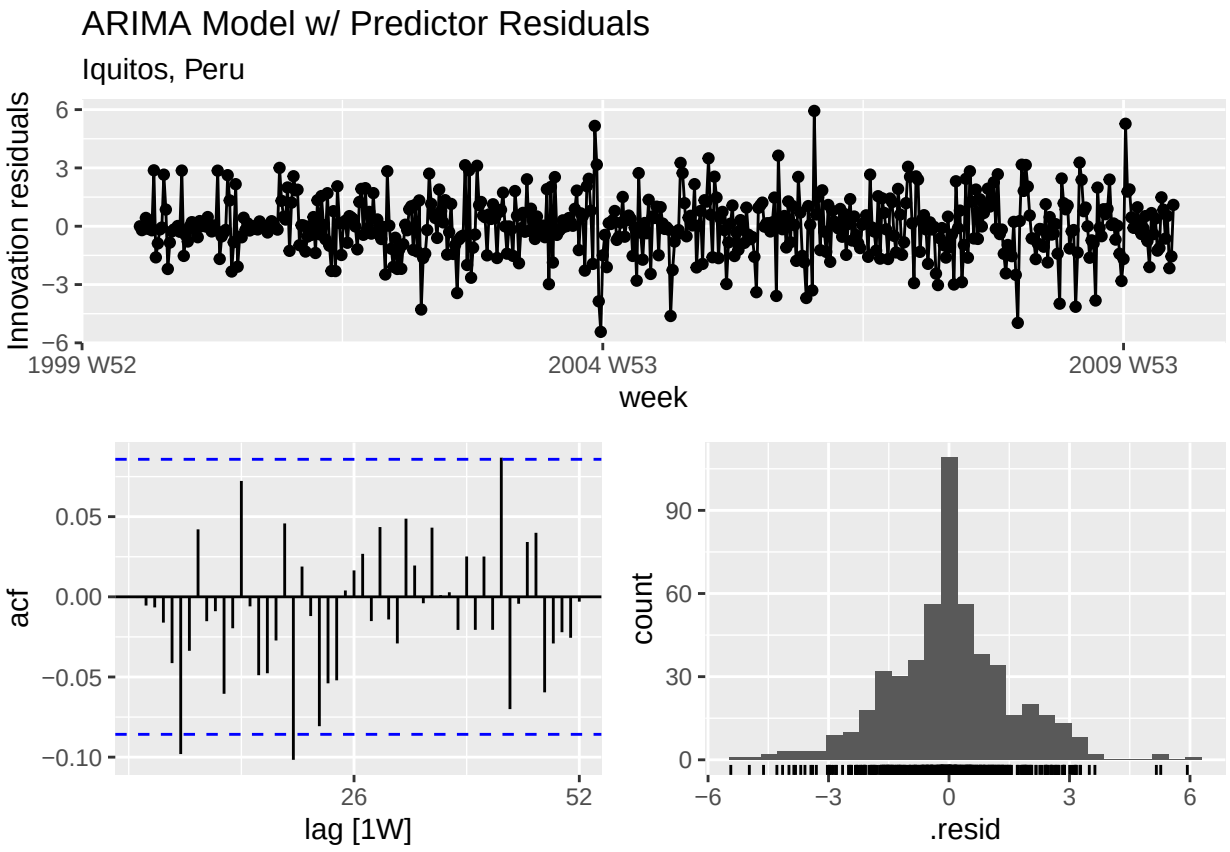
Precipitation shows weak correlation, but with slight increases as values are lagged. Since there is consistency across both cities I include the predictor. Additionally, the data are stationary across both cities. With that, the three predictors considered in the ARIMA are average temperature, dew point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

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))
```

```r
# Iquitos
dengue_iq_arima2 = dengue_iq_train %>%
  model(
    ARIMA(box_cox(total_cases, lambda) ~ lag(temp_diff) + lag(temp_diff, 2) +
            lag(dew_diff) + lag(dew_diff, 2) +
            lag(precip_diff) + lag(precip_diff, 2) +
            pdq(4,1,0) + PDQ(1,0,0))
  )

report(dengue_iq_arima2)
```

```
## Series: total_cases
## Model: LM w/ ARIMA(4,1,0)(1,0,0)[52] errors
## Transformation: box_cox(total_cases, lambda)
##
## Coefficients:
##           ar1      ar2      ar3      ar4    sar1  lag(temp_diff)
##       -0.3869  -0.1841  -0.0443  -0.0555  0.1437         -0.0337
## s.e.   0.0442   0.0477   0.0473   0.0435  0.0460          0.0494
##       lag(temp_diff, 2)  lag(dew_diff)  lag(dew_diff, 2)  lag(precip_diff)
##                 -0.0834         0.0405           -0.0231           -0.0028
## s.e.             0.0485         0.0558            0.0568            0.0015
##       lag(precip_diff, 2)
##                   -0.0032
## s.e.               0.0015
##
## sigma^2 estimated as 2.321:  log likelihood=-949.68
## AIC=1923.36   AICc=1923.97   BIC=1974.42
```

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




```
dengue_iq_arima2 %>%
  augment() %>%
  features(.innov, ljung_box, lag = 52)
```

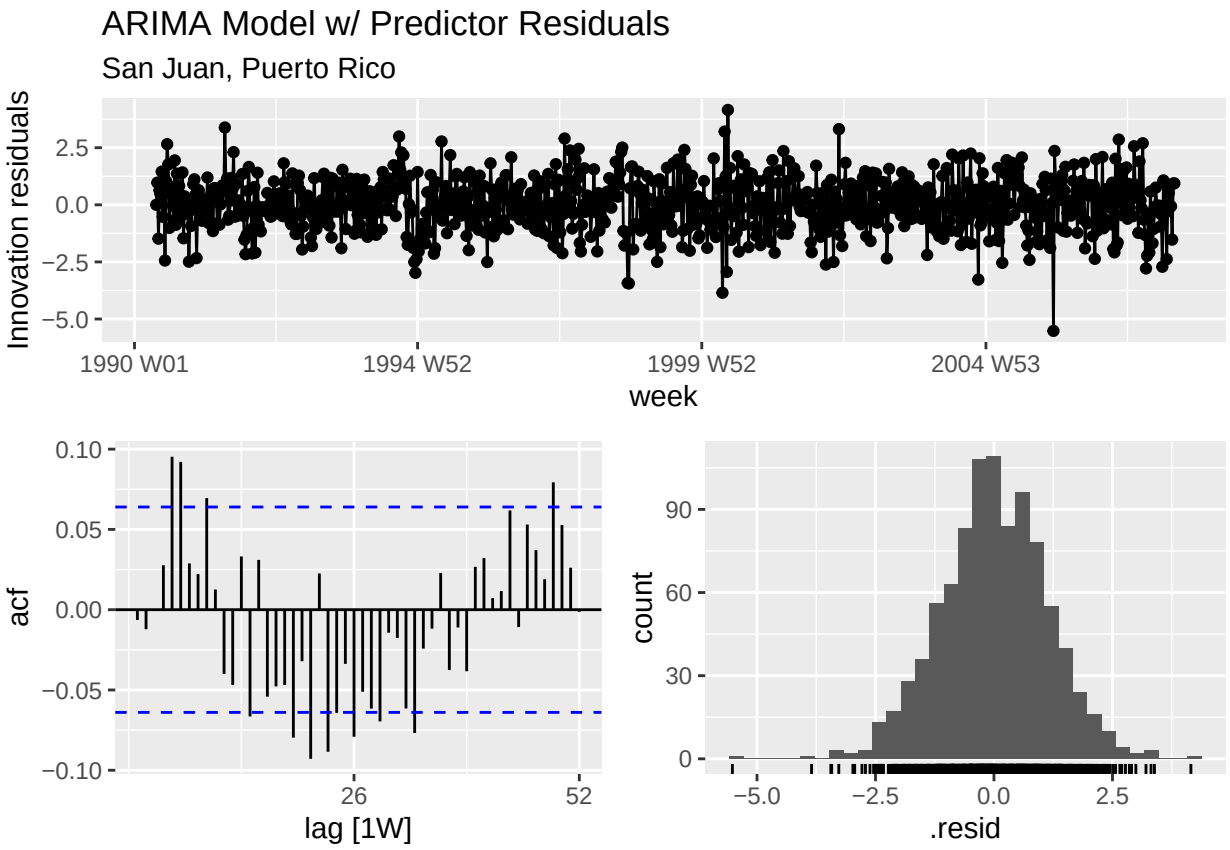
```
## # A tibble: 1 x 4
##   city  .model                                                  lb_stat lb_pvalue
##   <chr> <chr>                                                     <dbl>     <dbl>
## 1 iq    "ARIMA(box_cox(total_cases, lambda) ~ lag(temp_diff) ~    47.5     0.652
```

```
# San Juan
dengue_sj_arima2 = dengue_sj_train %>%
  model(
    ARIMA(box_cox(total_cases, lambda) ~ lag(temp_diff) + lag(temp_diff, 2) +
            lag(dew_diff) + lag(dew_diff, 2) +
            lag(precip_diff) + lag(precip_diff, 2) +
            pdq(4,1,0) + PDQ(1,0,0))
  )

report(dengue_sj_arima2)
```

```
## Series: total_cases
## Model: LM w/ ARIMA(4,1,0)(1,0,0)[52] errors
## Transformation: box_cox(total_cases, lambda)
##
## Coefficients:
```

```
##           ar1      ar2     ar3     ar4    sar1  lag(temp_diff)
##       -0.2702  -0.0927  0.0587  0.0839  0.0129          0.0909
## s.e.   0.0328   0.0343  0.0343  0.0335  0.0348          0.0679
##       lag(temp_diff, 2)  lag(dew_diff)  lag(dew_diff, 2)  lag(precip_diff)
##                  0.0486        -0.1183           -0.0864            0.0012
## s.e.             0.0679         0.0496            0.0491            0.0006
##       lag(precip_diff, 2)
##                    0.0014
## s.e.               0.0006
##
## sigma^2 estimated as 1.288:  log likelihood=-1440.97
## AIC=2905.95   AICc=2906.28   BIC=2964.07
```

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




```
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(box_cox(total_cases, lambda) ~ lag(temp_diff) ~    120.   3.02e-7
```

Neural Network

Prediction

Limitations

The biggest limitation of the TSLM and Neural Network 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.

Future Work

Conclusions

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

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