Data Exlporation

Jenn Havens and Madison Hobbs

November 8, 2017

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Loading required package: rpart

## Loading required package: caret

## Loading required package: lattice

## Loading required package: tidyverse

## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
## lift():   purrr, caret

## Loading required package: ggcorrplot
```

Variable Selection

Variable descriptions https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/page/82/.

Week and week of year provide the same information, so we'll keep only week inside the model.

There are three measures of precipitation. One is station_precip_mm which is the total daily precipitation as measured by NOAA's GHCN weather stations (https://www.ncdc.noaa.gov/ghcn-daily-description). Another is precipitation_amt_mm which represents total precipitation as measured by PERSIANN satellites. The third and forth, reanalysis_sat_precip_amt_mm and reanalysis_precip_amt_kg_per_m2 are both generated by NOAA's NCEP Climate Forecast System Reanalysis (https://climatedataguide.ucar.edu/climate-data/climate-forecast-system-reanalysis-cfsr).

We should choose one of these precipitation measures for the model, since these four precipitation measures all measure approximately the same thing. We decide on the daily precipitation as measured by NOAA's

GHCN (https://www.ncdc.noaa.gov/ghcn-daily-description) because it represents actually recorded daily measurements, not satellite or model estimates. It is as close to the source as we can get.

Same thing for minimum temperature, maximum temperature, average temperature, and diurnal temperature.

```
data_for_model <- TrainFull %>% select(city, year, weekofyear, total_cases, station_precip_mm, station_a
```

Which Air Temperature Measures Should Be Included?

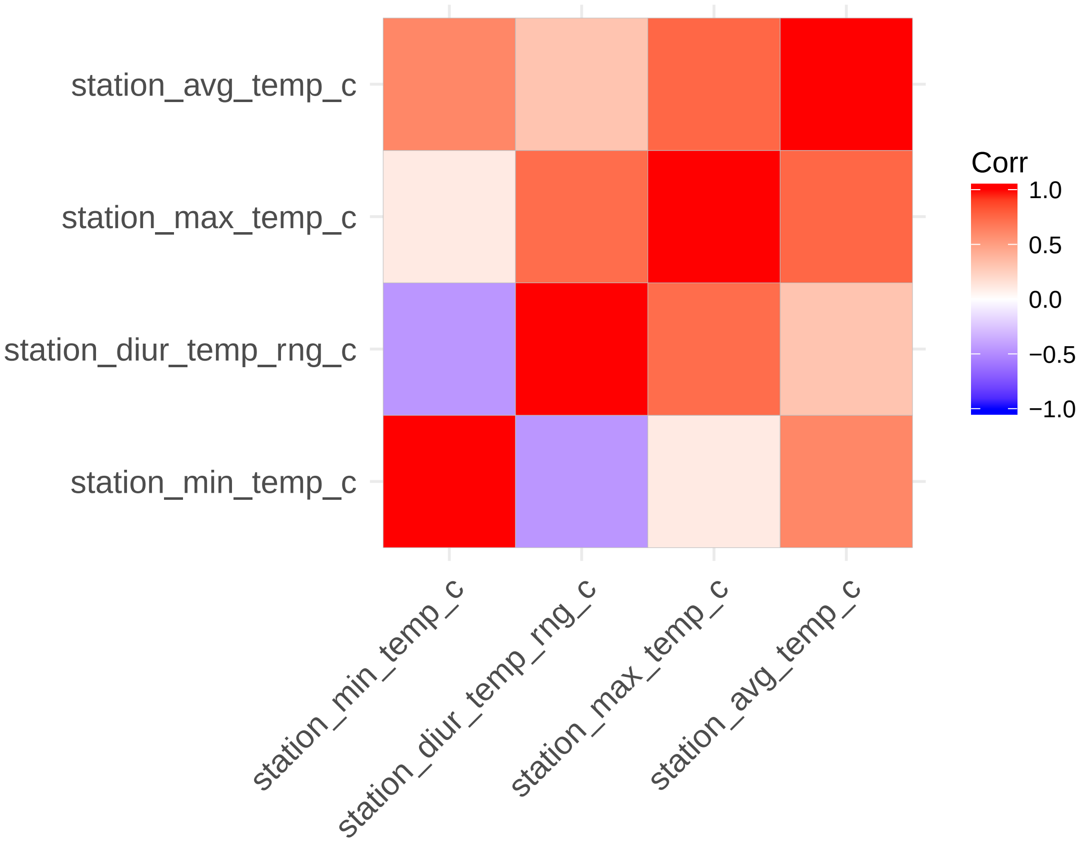
Note that Choi et. al say : "Mean temperature was significantly associated with dengue incidence in all three provinces, but incidence did not correlate well with maximum temperature in Banteay Meanchey, nor with minimum temperature in Kampong Thom at a lag of three months in the negative binomial model." I'm inclined to only use mean temperature.

```
temps <- TrainFull %>% select(station_max_temp_c,station_avg_temp_c, station_min_temp_c, station_diur_te

corr_matrix <- cor(temps)
corr_pmat <- cor_pmat(temps)
# they are all significantly correlated
corr_pmat < 0.05
```

```
##                         station_max_temp_c station_avg_temp_c
## station_max_temp_c                    TRUE               TRUE
## station_avg_temp_c                    TRUE               TRUE
## station_min_temp_c                    TRUE               TRUE
## station_diur_temp_rng_c               TRUE               TRUE
##                         station_min_temp_c station_diur_temp_rng_c
## station_max_temp_c                    TRUE                    TRUE
## station_avg_temp_c                    TRUE                    TRUE
## station_min_temp_c                    TRUE                    TRUE
## station_diur_temp_rng_c               TRUE                    TRUE
```

```
ggcorrplot(corr_matrix, hc.order = TRUE)
```



Should humidity and Dewpoint be included?

There are a series of variables generated by the Climate Forecast System Reanalysis on a 0.5 by 0.5 degree scale. The estimate precipitation, dew point, air temperature (actual, max, min, avg), relative and specific humidity, and diurnal temperature range.

Creating time lag variabes

They found significant association between three month lag and prediciting dengue cases. So we'll move by the 12 weeks before each observation.

1) Precipitation

```
precip_lag <- c()

sj_precip <- data_for_model %>% filter(city == "sj") %>% select(station_precip_mm)
sj_precip <- sj_precip$station_precip_mm

index = 12
while(index < length(sj_precip)) {
  precip_lag <- c(precip_lag, mean(sj_precip[index-11:index]))
  index <- index + 1
}

precip_lag <- c(mean(sj_precip[1:11]), precip_lag)
precip_lag <- c(mean(sj_precip[1:10]), precip_lag)
```

```r
precip_lag <- c(mean(sj_precip[1:9]), precip_lag)
precip_lag <- c(mean(sj_precip[1:8]), precip_lag)
precip_lag <- c(mean(sj_precip[1:7]), precip_lag)
precip_lag <- c(mean(sj_precip[1:6]), precip_lag)
precip_lag <- c(mean(sj_precip[1:5]), precip_lag)
precip_lag <- c(mean(sj_precip[1:4]), precip_lag)
precip_lag <- c(mean(sj_precip[1:3]), precip_lag)
precip_lag <- c(mean(sj_precip[1:2]), precip_lag)
precip_lag <- c(mean(sj_precip[1]), precip_lag)
precip_lag <- c(mean(sj_precip[1]), precip_lag)

sj_data_for_model <- data_for_model %>% filter(city == "sj") %>% mutate(precip_lag = precip_lag)
```

Building a Model

The Driven Data team uses Mean Absolute Error to measure error, so we will, too.

```r
# returns Mean Absolute Error
# error is defined as actual - predicted
MAE <- function(error)
{
    mean(abs(error))
}

# returns Root Mean Squared Error
# error is defined as actual - predicted
RMSE <- function(error)
{
  sqrt(mean(error^2))
}
```

```r
# we should remove the san juan data for which we don't have true precip or temp lag
sj_data_for_model <- sj_data_for_model %>% slice(13:n())

# get out test set
set.seed(48)
sj_test <- sample_n(sj_data_for_model, 71)
sj_train <- anti_join(sj_data_for_model, sj_test)
```

```
## Joining, by = c("city", "year", "weekofyear", "total_cases", "station_precip_mm", "station_avg_temp_
```

```r
sj_rf_train <- train(total_cases ~ ., data = sj_train, method = "rf", trControl = trainControl(method =

sj_rf_train
```

```
## Random Forest
##
## 644 samples
##   6 predictor
##
## No pre-processing
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared
##   1     29.42287  0.3460674
##   2     22.92806  0.6029021
##   3     21.28141  0.6578916
##   4     21.14889  0.6621390
##   5     21.00591  0.6666919
##   6     21.23771  0.6592951
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was mtry = 5.
```

```
varImp(sj_rf_train)
```

```
## rf variable importance
##
##                     Overall
## weekofyear         100.0000
## year                82.2878
## precip_lag          76.7383
## station_avg_temp_c  14.4721
## station_precip_mm    0.5499
## citysj               0.0000
```

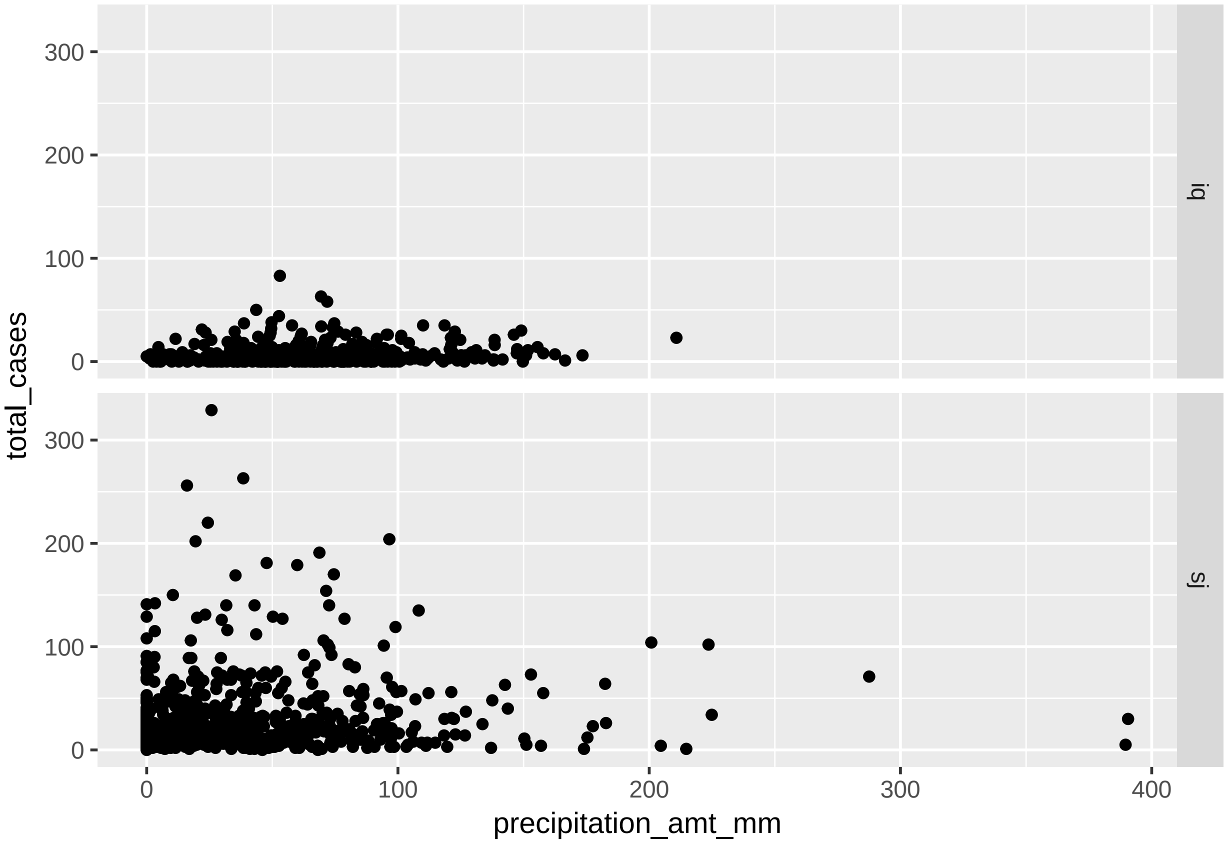
```
preds <- predict(sj_rf_train, sj_test)
MAE(sj_test$total_cases - preds)
```

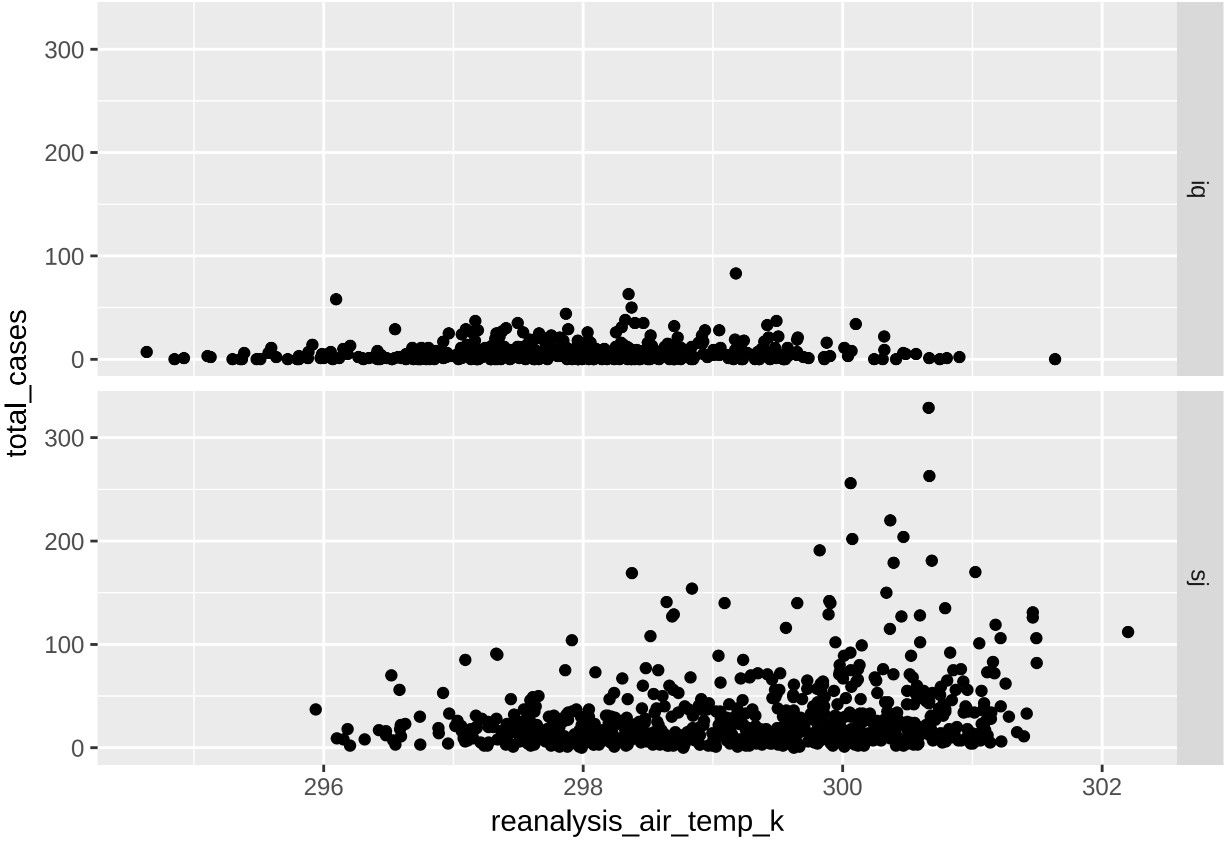
```
## [1] 10.215
```

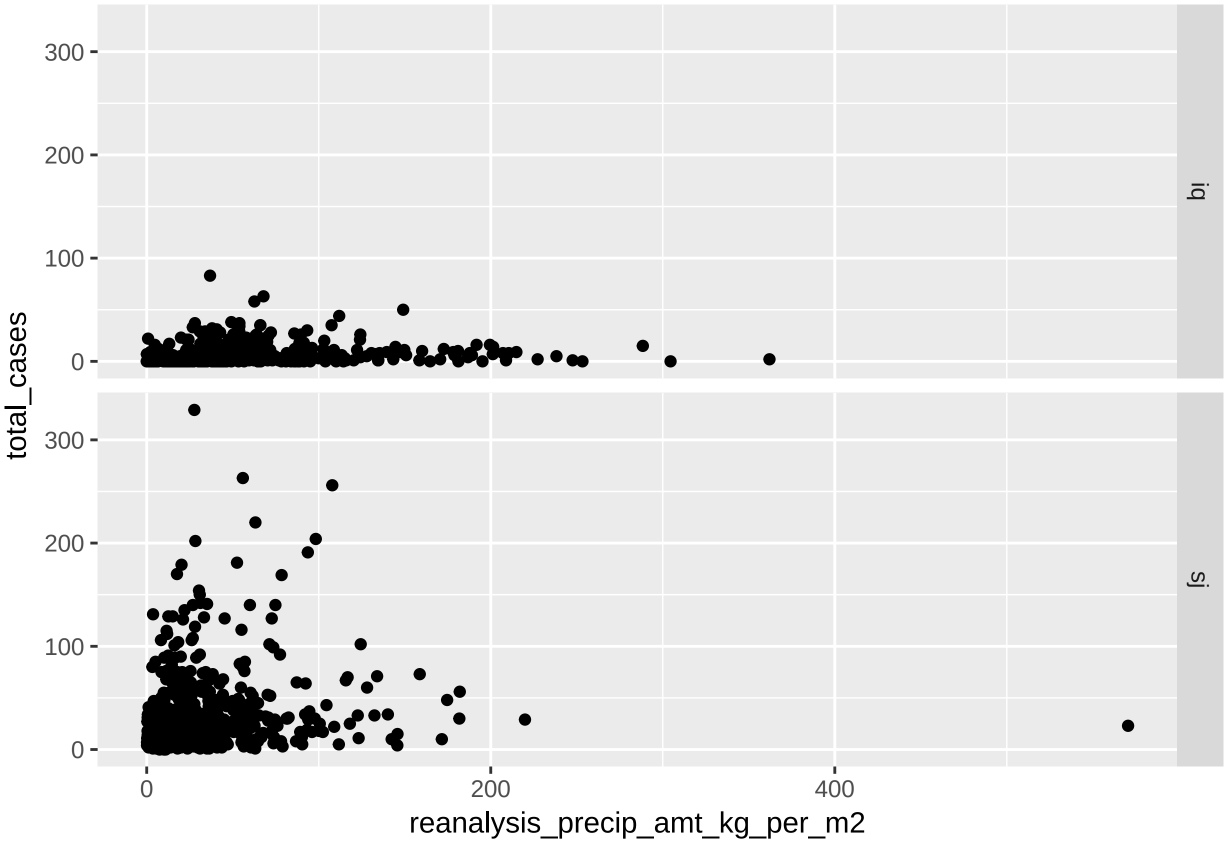
```
RMSE(sj_test$total_cases - preds)
```

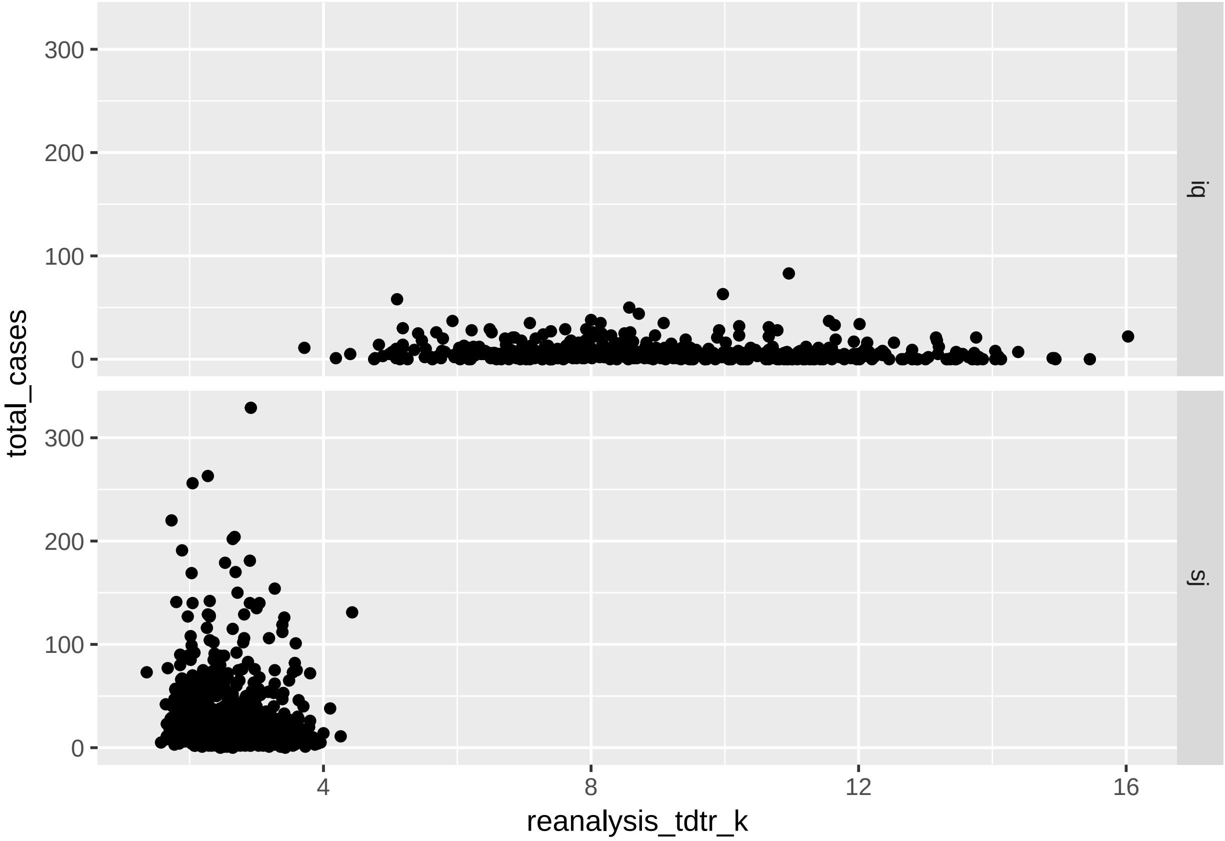
```
## [1] 18.93989
```

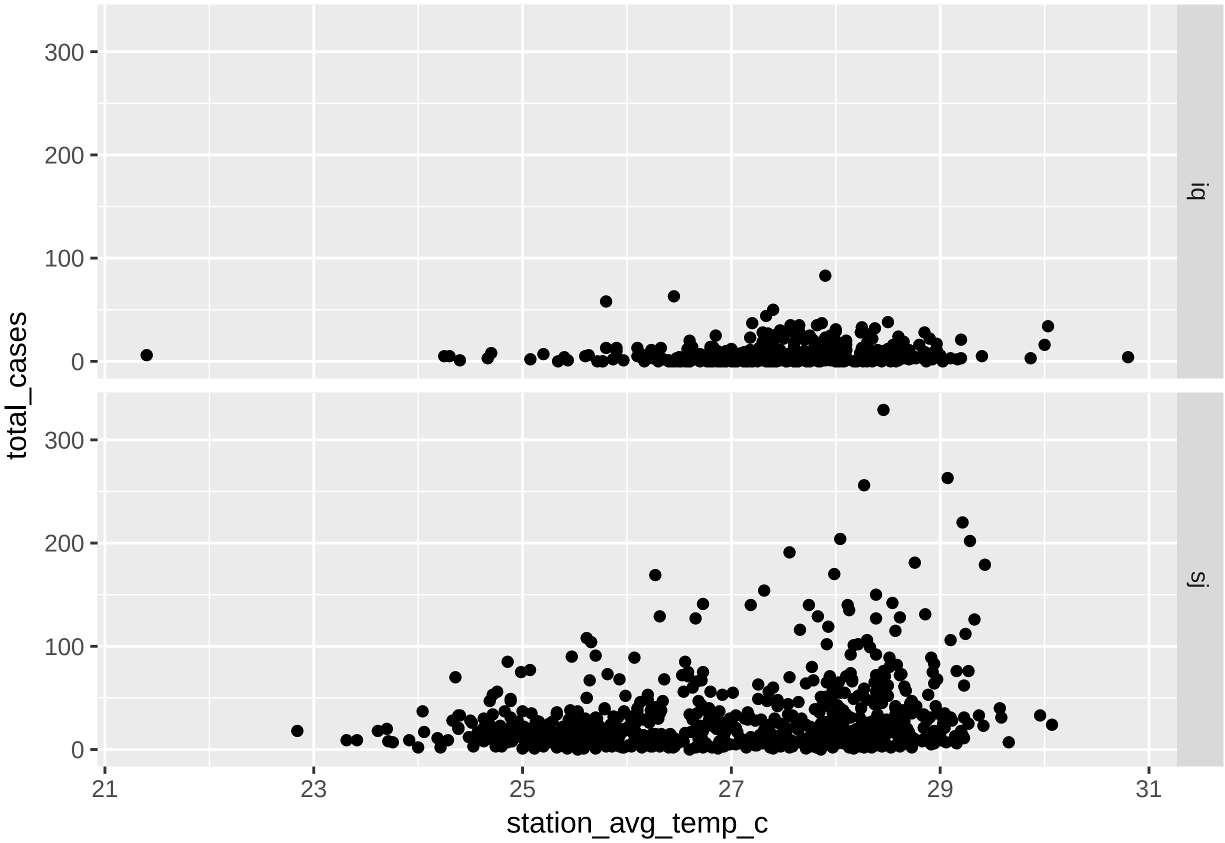
For a random forest with only city, year, week of year, station_precip_mm, and station_avg_temp_c, and precip_lag we get MAE =10.2 and RMSE = 18.9.

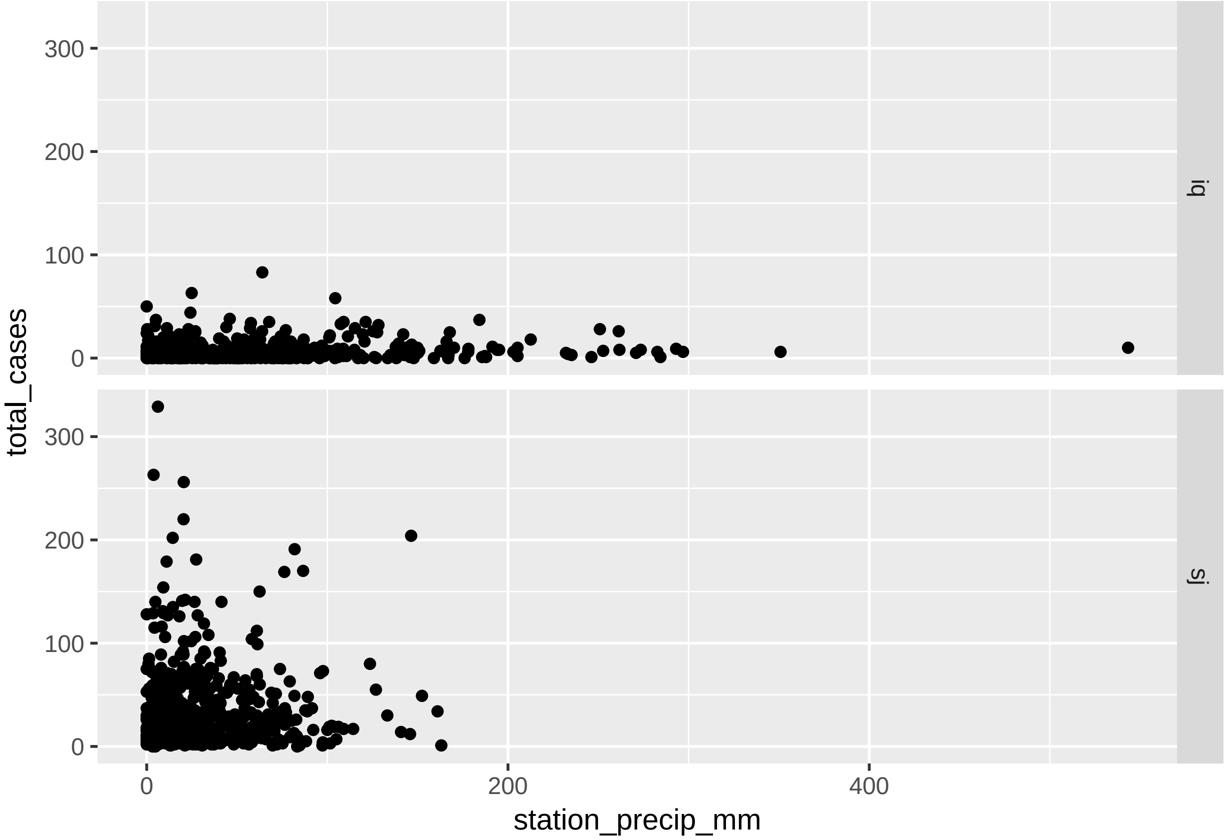

total_cases
300
200
100
0
iq
sj
0
100
200
300
400
precipitation_amt_mm

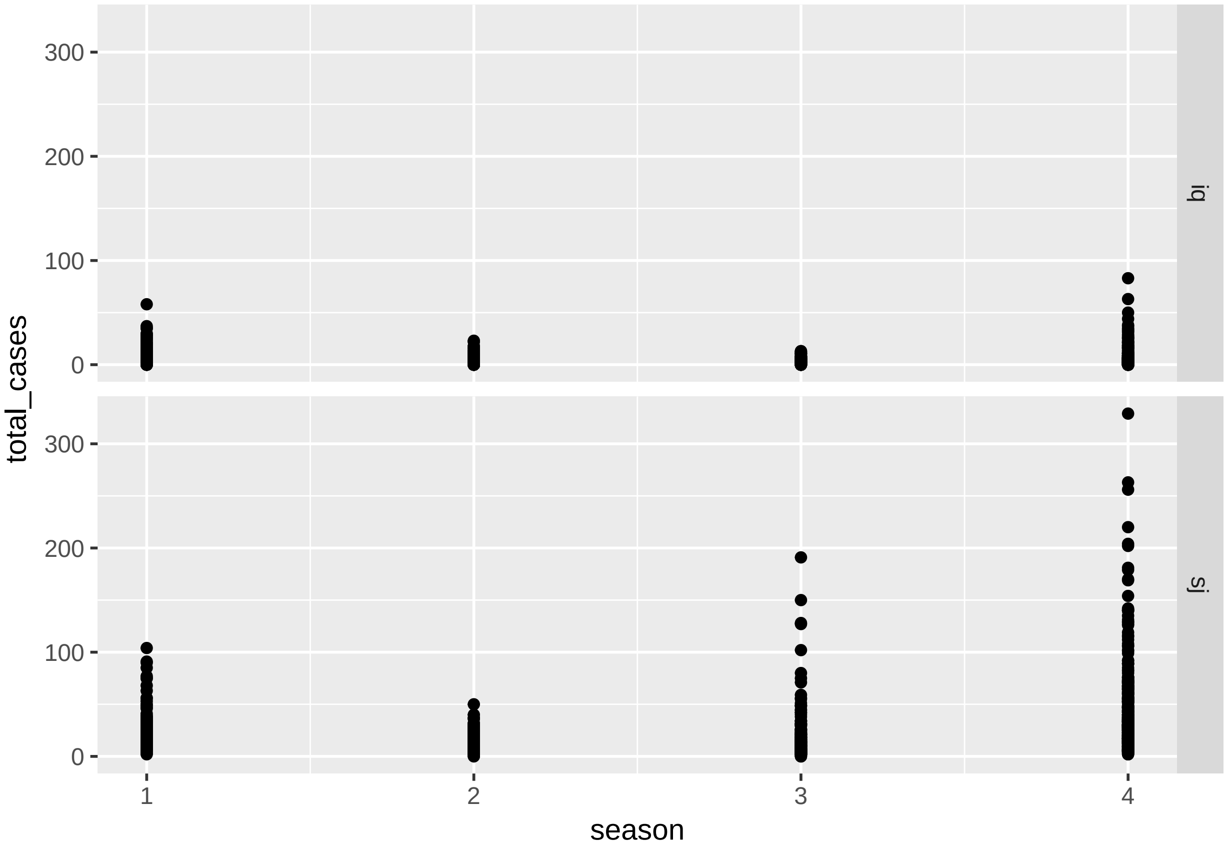

total_cases
300
200
100
0
iq
sj
296
298
300
302
reanalysis_air_temp_k

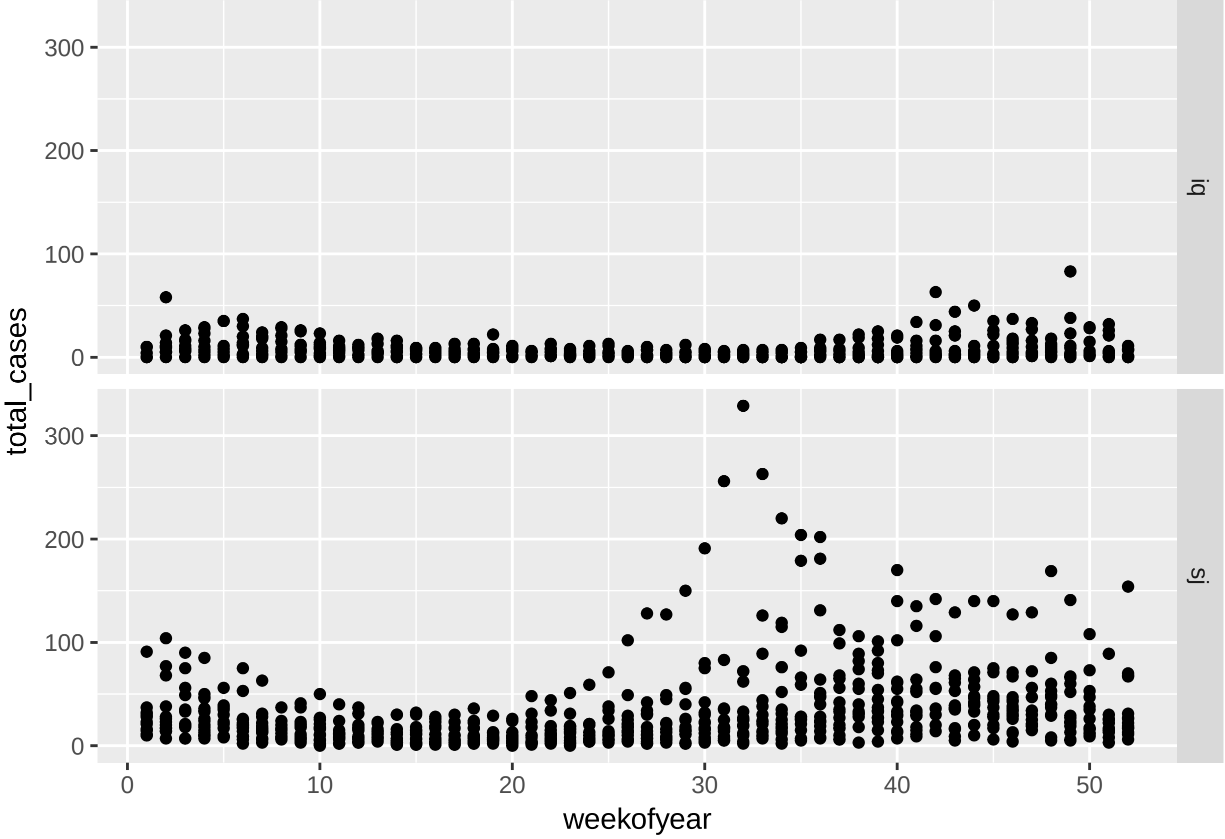

total_cases
300
200
100
0
iq
sj
0
200
400
reanalysis_precip_amt_kg_per_m2

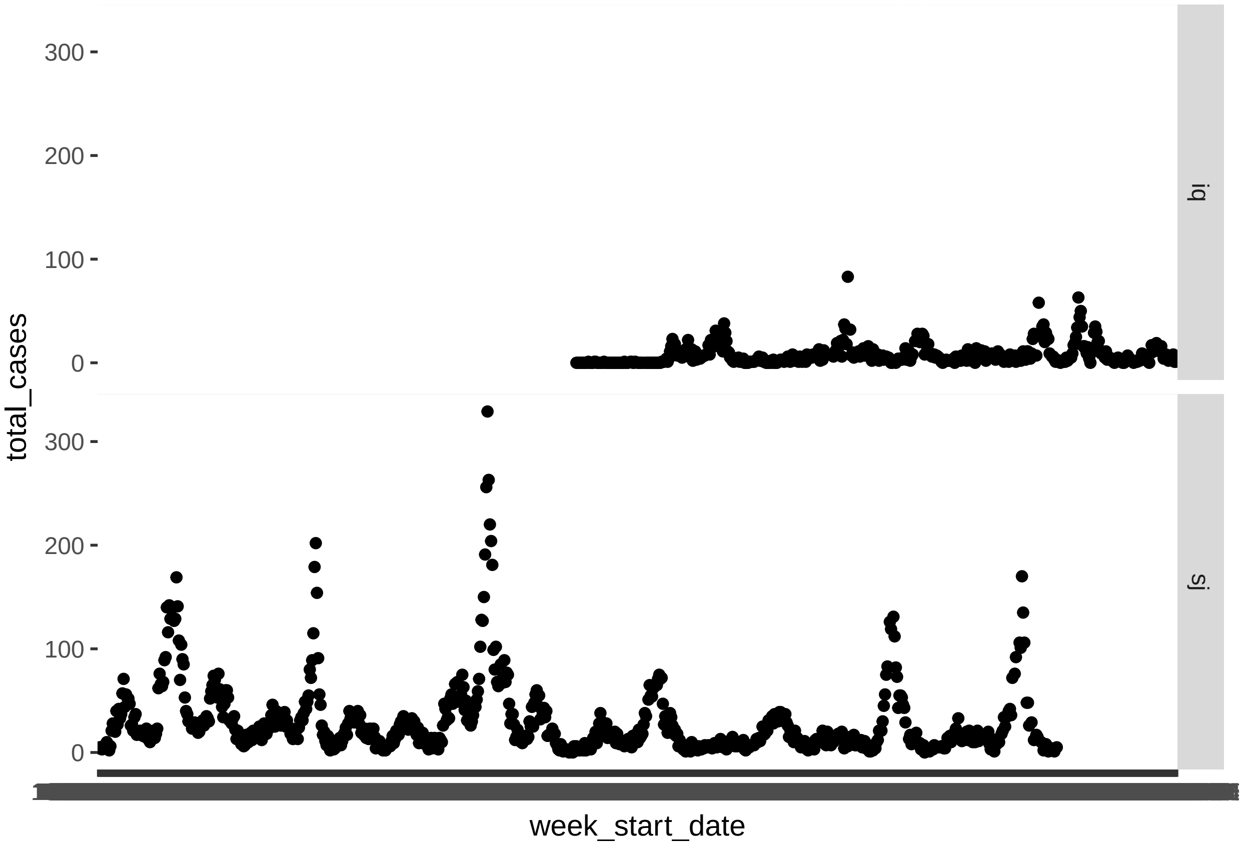

total_cases
300
200
100
0
iq
sj
4
8
12
16
reanalysis_tdtr_k


total_cases
300
200
100
0
iq
sj
21
23
25
27
29
31
station_avg_temp_c


total_cases
300
200
100
0
iq
sj
0
200
400
station_precip_mm


total_cases
300
200
100
0
iq
sj
1
2
3
4
season


total_cases
300
200
100
0
iq
sj
0
10
20
30
40
50
weekofyear


300
200
100
0
300
200
100
0
total_cases
iq
sj
week_start_date