A

Group Project Report

on

DengAI: Predicting Disease Spread

By

Group -3 (Team Y)

Chandu Yerragopu

Saran Prasad Balasubramaniam

Wen Xie

Under the esteemed guidance of
Dr. Paige Rutner, Ph.D.
Course: ISQS 6349 Predictive Analytics

TEXAS TECH UNIVERSITY
Rawls College *of* Business™

Table of contents

1 Introduction

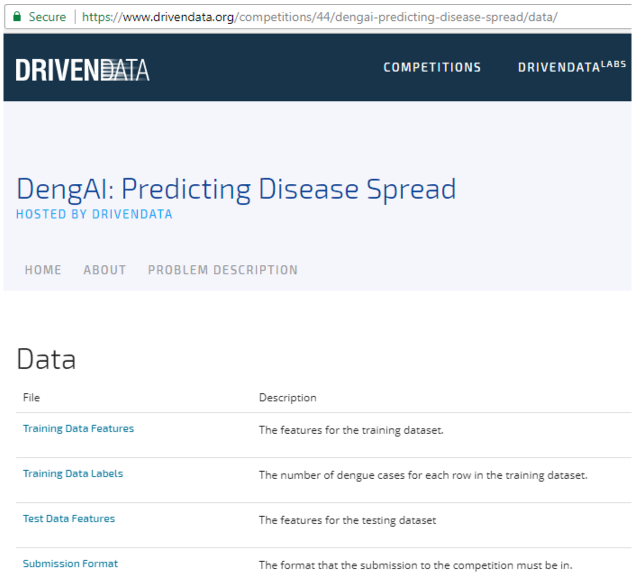
Dengue fever is a mosquito-borne disease that occurs in tropical and sub-tropical parts of the world. In mild cases, symptoms are similar to the flu: fever, rash, and muscle and joint pain. In severe cases, dengue fever can cause severe bleeding, low blood pressure, and even death.

Because it is carried by mosquitoes, the transmission dynamics of dengue are related to climate variables such as temperature and precipitation. Although the relationship to climate is complex, a growing number of scientists argue that climate change is likely to produce distributional shifts that will have significant public health implications worldwide. In recent years dengue fever has been spreading. Historically, the disease has been most prevalent in Southeast Asia and the Pacific islands. These days many of the nearly half billion cases per year are occurring in Latin America.

1.1 Dataset source

Using environmental data collected by various U.S. Federal Government agencies—from the Centers for Disease Control and Prevention to the National Oceanic and Atmospheric Administration in the U.S. Department of Commerce. Below is the competition site where data is downloaded from:

https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/data/


Secure
https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/data/
DRIVENDATA
COMPETITIONS
DRIVENDATA LABS
HOME
ABOUT
PROBLEM DESCRIPTION

DengAI: Predicting Disease Spread

HOSTED BY DRIVENDATA

Data

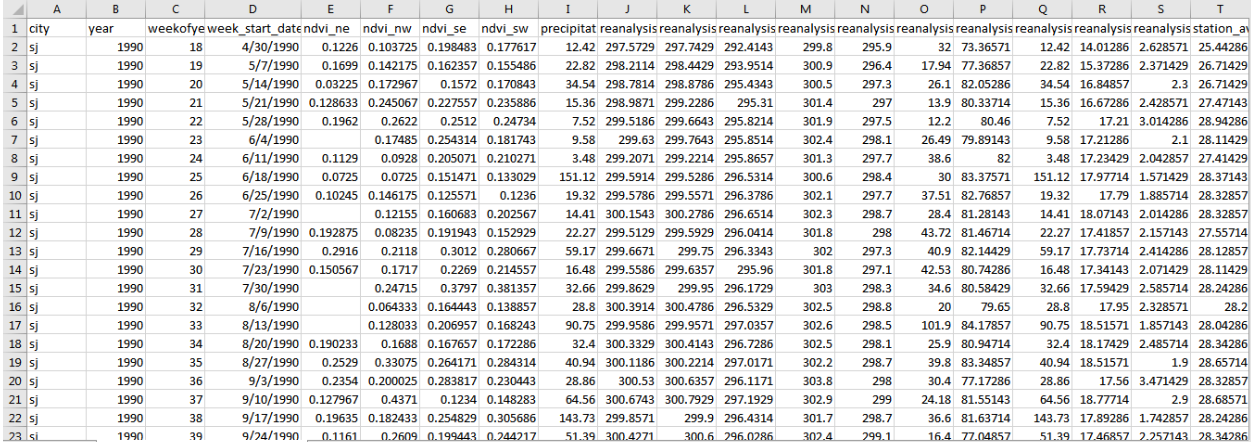
File	Description
Training Data Features	The features for the training dataset.
Training Data Labels	The number of dengue cases for each row in the training dataset.
Test Data Features	The features for the testing dataset
Submission Format	The format that the submission to the competition must be in.

1.2 How data looks

From the above screenshot, we could see that we have the following files and also how the data looks like

1. DengAI_Predicting_Disease_Spread_-_Training_Data_Features

   This dataset contains information on 1456 weeks for the 2 cities across 24 features.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	city	year	weekofye	week_start_date	ndvi_ne	ndvi_nw	ndvi_se	ndvi_sw	precipitat	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	station_av
2	sj	1990	18	4/30/1990	0.1226	0.103725	0.198483	0.177617	12.42	297.5729	297.7429	292.4143	299.8	295.9	32	73.36571	12.42	14.01286	2.628571	25.44286
3	sj	1990	19	5/7/1990	0.1699	0.142175	0.162357	0.155486	22.82	298.2114	298.4429	293.9514	300.9	296.4	17.94	77.36857	22.82	15.37286	2.371429	26.71429
4	sj	1990	20	5/14/1990	0.03225	0.172967	0.1572	0.170843	34.54	298.7814	298.8786	295.4343	300.5	297.3	26.1	82.05286	34.54	16.84857	2.3	26.71429
5	sj	1990	21	5/21/1990	0.128633	0.245067	0.227557	0.235886	15.36	298.9871	299.2286	295.31	301.4	297	13.9	80.33714	15.36	16.67286	2.428571	27.47143
6	sj	1990	22	5/28/1990	0.1962	0.2622	0.2512	0.24734	7.52	299.5186	299.6643	295.8214	301.9	297.5	12.2	80.46	7.52	17.21	3.014286	28.94286
7	sj	1990	23	6/4/1990		0.17485	0.254314	0.181743	9.58	299.63	299.7643	295.8514	302.4	298.1	26.49	79.89143	9.58	17.21286	2.1	28.11429
8	sj	1990	24	6/11/1990	0.1129	0.0928	0.205071	0.210271	3.48	299.2071	299.2214	295.8657	301.3	297.7	38.6	82	3.48	17.23429	2.042857	27.41429
9	sj	1990	25	6/18/1990	0.0725	0.0725	0.151471	0.133029	151.12	299.5914	299.5286	296.5314	300.6	298.4	30	83.37571	151.12	17.97714	1.571429	28.37143
10	sj	1990	26	6/25/1990	0.10245	0.146175	0.125571	0.1236	19.32	299.5786	299.5571	296.3786	302.1	297.7	37.51	82.76857	19.32	17.79	1.885714	28.32857
11	sj	1990	27	7/2/1990		0.12155	0.160683	0.202567	14.41	300.1543	300.2786	296.6514	302.3	298.7	28.4	81.28143	14.41	18.07143	2.014286	28.32857
12	sj	1990	28	7/9/1990	0.192875	0.08235	0.191943	0.152929	22.27	299.5129	299.5929	296.0414	301.8	298	43.72	81.46714	22.27	17.41857	2.157143	27.55714
13	sj	1990	29	7/16/1990	0.2916	0.2118	0.3012	0.280667	59.17	299.6671	299.75	296.3343	302	297.3	40.9	82.14429	59.17	17.73714	2.414286	28.12857
14	sj	1990	30	7/23/1990	0.150567	0.1717	0.2269	0.214557	16.48	299.5586	299.6357	295.96	301.8	297.1	42.53	80.74286	16.48	17.34143	2.071429	28.11429
15	sj	1990	31	7/30/1990		0.24715	0.3797	0.381357	32.66	299.8629	299.95	296.1729	303	298.3	34.6	80.58429	32.66	17.59429	2.585714	28.24286
16	sj	1990	32	8/6/1990		0.064333	0.164443	0.138857	28.8	300.3914	300.4786	296.5329	302.5	298.8	20	79.65	28.8	17.95	2.328571	28.2
17	sj	1990	33	8/13/1990		0.128033	0.206957	0.168243	90.75	299.9586	299.9571	297.0357	302.6	298.5	101.9	84.17857	90.75	18.51571	1.857143	28.04286
18	sj	1990	34	8/20/1990	0.190233	0.1688	0.167657	0.172286	32.4	300.3329	300.4143	296.7286	302.5	298.1	25.9	80.94714	32.4	18.17429	2.485714	28.34286
19	sj	1990	35	8/27/1990	0.2529	0.33075	0.264171	0.284314	40.94	300.1186	300.2214	297.0171	302.2	298.7	39.8	83.34857	40.94	18.51571	1.9	28.65714
20	sj	1990	36	9/3/1990	0.2354	0.200025	0.283817	0.230443	28.86	300.53	300.6357	296.1171	303.8	298	30.4	77.17286	28.86	17.56	3.471429	28.32857
21	sj	1990	37	9/10/1990	0.127967	0.4371	0.1234	0.148283	64.56	300.6743	300.7929	297.1929	302.9	299	24.18	81.55143	64.56	18.77714	2.9	28.68571
22	sj	1990	38	9/17/1990	0.19635	0.182433	0.254829	0.305686	143.73	299.8571	299.9	296.4314	301.7	298.7	36.6	81.63714	143.73	17.89286	1.742857	28.24286
23	sj	1990	39	9/24/1990	0.1161	0.2609	0.199443	0.244217	51.39	300.4271	300.6	296.0286	302.4	299.1	16.4	77.04857	51.39	17.46857	2.257143	28.34286

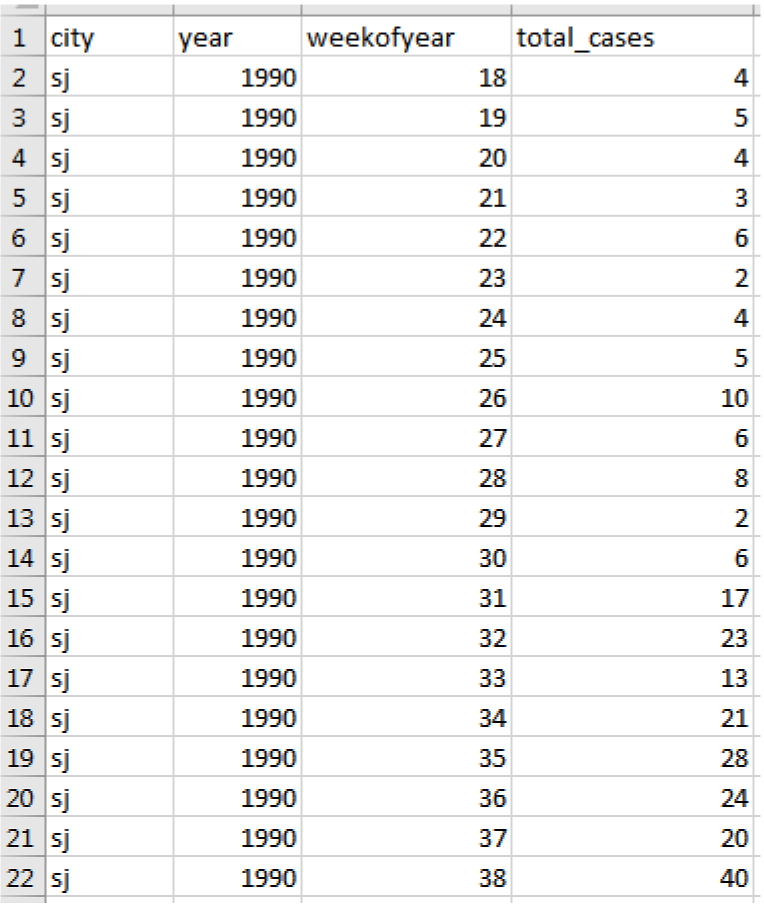
The variables in the data are as follows:

city, year, weekofyear,week_start_date,ndvi_ne,ndvi_nw, ndvi_se,ndvi_sw, precipitation_amt_mm, reanalysis_air_temp_k,reanalysis_avg_temp_k, reanalysis_dew_point_temp_k, reanalysis_max_air_temp_k, reanalysis_min_air_temp_k, reanalysis_precip_amt_kg_per_m2, reanalysis_relative_humidity_percent, reanalysis_sat_precip_amt_mm, reanalysis_specific_humidity_g_per_kg, reanalysis_tdtr_k,station_avg_temp_c, station_diur_temp_rng_c,station_max_temp_c, station_min_temp_c, station_precip_mm

Then we have the Data labels as shown below:

2. DengAI_Predicting_Disease_Spread_-_Training_Data_Labels

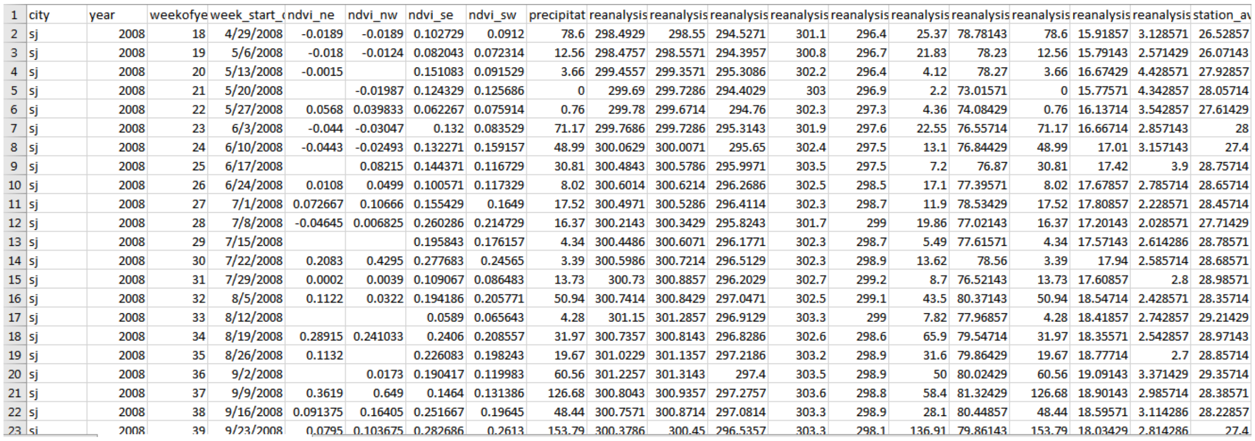
It contains the same fields of above Training_data_features and also lists the total_cases which is the value that we have to predict using the test data.



1	city	year	weekofyear	total_cases
2	sj	1990	18	4
3	sj	1990	19	5
4	sj	1990	20	4
5	sj	1990	21	3
6	sj	1990	22	6
7	sj	1990	23	2
8	sj	1990	24	4
9	sj	1990	25	5
10	sj	1990	26	10
11	sj	1990	27	6
12	sj	1990	28	8
13	sj	1990	29	2
14	sj	1990	30	6
15	sj	1990	31	17
16	sj	1990	32	23
17	sj	1990	33	13
18	sj	1990	34	21
19	sj	1990	35	28
20	sj	1990	36	24
21	sj	1990	37	20
22	sj	1990	38	40

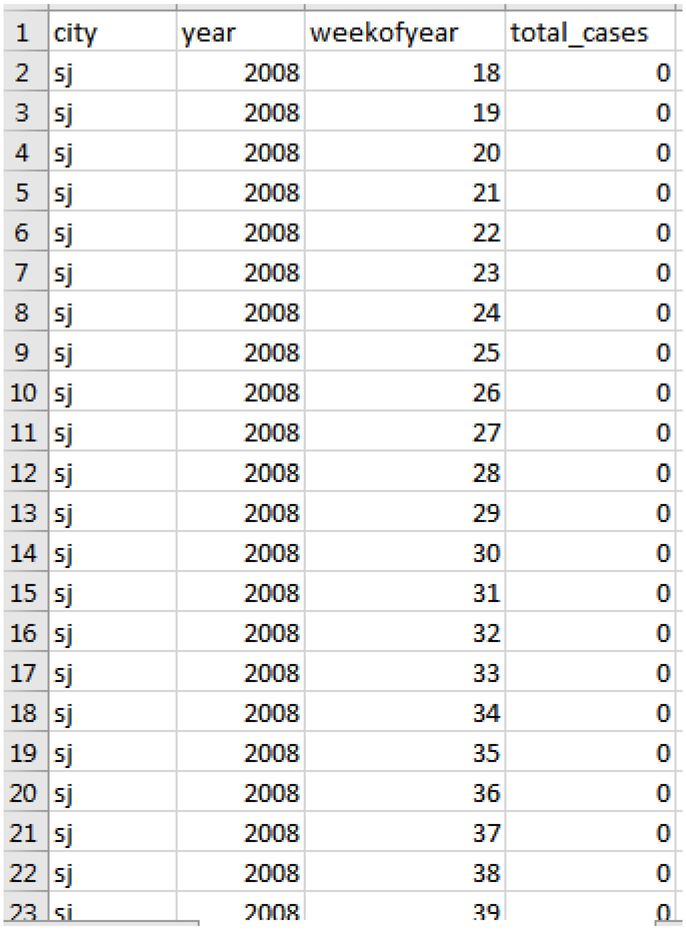
3. DengAI_Predicting_Disease_Spread_-_Test_Data_Features

This data has the same set of columns as in the Training data and we need to use it for testing our predictive model.



1	city	year	weekofye	week_start_(	ndvi_ne	ndvi_nw	ndvi_se	ndvi_sw	precipitat	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	reanalysis	station_av
2	sj	2008	18	4/29/2008	-0.0189	-0.0189	0.102729	0.0912	78.6	298.4929	298.55	294.5271	301.1	296.4	25.37	78.78143	78.6	15.91857	3.128571	26.52857
3	sj	2008	19	5/6/2008	-0.018	-0.0124	0.082043	0.072314	12.56	298.4757	298.5571	294.3957	300.8	296.7	21.83	78.23	12.56	15.79143	2.571429	26.07143
4	sj	2008	20	5/13/2008	-0.0015		0.151083	0.091529	3.66	299.4557	299.3571	295.3086	302.2	296.4	4.12	78.27	3.66	16.67429	4.428571	27.92857
5	sj	2008	21	5/20/2008		-0.01987	0.124329	0.125686	0	299.69	299.7286	294.4029	303	296.9	2.2	73.01571	0	15.77571	4.342857	28.05714
6	sj	2008	22	5/27/2008	0.0568	0.039833	0.062267	0.075914	0.76	299.78	299.6714	294.76	302.3	297.3	4.36	74.08429	0.76	16.13714	3.542857	27.61429
7	sj	2008	23	6/3/2008	-0.044	-0.03047	0.132	0.083529	71.17	299.7686	299.7286	295.3143	301.9	297.6	22.55	76.55714	71.17	16.66714	2.857143	28
8	sj	2008	24	6/10/2008	-0.0443	-0.02493	0.132271	0.159157	48.99	300.0629	300.0071	295.65	302.4	297.5	13.1	76.84429	48.99	17.01	3.157143	27.4
9	sj	2008	25	6/17/2008		0.08215	0.144371	0.116729	30.81	300.4843	300.5786	295.9971	303.5	297.5	7.2	76.87	30.81	17.42	3.9	28.75714
10	sj	2008	26	6/24/2008	0.0108	0.0499	0.100571	0.117329	8.02	300.6014	300.6214	296.2686	302.5	298.5	17.1	77.39571	8.02	17.67857	2.785714	28.65714
11	sj	2008	27	7/1/2008	0.072667	0.10666	0.155429	0.1649	17.52	300.4971	300.5286	296.4114	302.3	298.7	11.9	78.53429	17.52	17.80857	2.228571	28.45714
12	sj	2008	28	7/8/2008	-0.04645	0.006825	0.260286	0.214729	16.37	300.2143	300.3429	295.8243	301.7	299	19.86	77.02143	16.37	17.20143	2.028571	27.71429
13	sj	2008	29	7/15/2008			0.195843	0.176157	4.34	300.4486	300.6071	296.1771	302.3	298.7	5.49	77.61571	4.34	17.57143	2.614286	28.78571
14	sj	2008	30	7/22/2008	0.2083	0.4295	0.277683	0.24565	3.39	300.5986	300.7214	296.5129	302.3	298.9	13.62	78.56	3.39	17.94	2.585714	28.68571
15	sj	2008	31	7/29/2008	0.0002	0.0039	0.109067	0.086483	13.73	300.73	300.8857	296.2029	302.7	299.2	8.7	76.52143	13.73	17.60857	2.8	28.98571
16	sj	2008	32	8/5/2008	0.1122	0.0322	0.194186	0.205771	50.94	300.7414	300.8429	297.0471	302.5	299.1	43.5	80.37143	50.94	18.54714	2.428571	28.35714
17	sj	2008	33	8/12/2008			0.0589	0.065643	4.28	301.15	301.2857	296.9129	303.3	299	7.82	77.96857	4.28	18.41857	2.742857	29.21429
18	sj	2008	34	8/19/2008	0.28915	0.241033	0.2406	0.208557	31.97	300.7357	300.8143	296.8286	302.6	298.6	65.9	79.54714	31.97	18.35571	2.542857	28.97143
19	sj	2008	35	8/26/2008	0.1132		0.226083	0.198243	19.67	301.0229	301.1357	297.2186	303.2	298.9	31.6	79.86429	19.67	18.77714	2.7	28.85714
20	sj	2008	36	9/2/2008		0.0173	0.190417	0.119983	60.56	301.2257	301.3143	297.4	303.5	298.9	50	80.02429	60.56	19.09143	3.371429	29.35714
21	sj	2008	37	9/9/2008	0.3619	0.649	0.1464	0.131386	126.68	300.8043	300.9357	297.2757	303.6	298.8	58.4	81.32429	126.68	18.90143	2.985714	28.38571
22	sj	2008	38	9/16/2008	0.091375	0.16405	0.251667	0.19645	48.44	300.7571	300.8714	297.0814	303.3	298.9	28.1	80.44857	48.44	18.59571	3.114286	28.22857
23	sj	2008	39	9/23/2008	0.0795	0.103675	0.282686	0.2613	153.79	300.3786	300.45	296.5357	303.3	298.1	136.91	79.86143	153.79	18.03429	2.814286	27.4

4. DengAI_Predicting_Disease_Spread_-_Submission_Format



1	city	year	weekofyear	total_cases
2	sj	2008	18	0
3	sj	2008	19	0
4	sj	2008	20	0
5	sj	2008	21	0
6	sj	2008	22	0
7	sj	2008	23	0
8	sj	2008	24	0
9	sj	2008	25	0
10	sj	2008	26	0
11	sj	2008	27	0
12	sj	2008	28	0
13	sj	2008	29	0
14	sj	2008	30	0
15	sj	2008	31	0
16	sj	2008	32	0
17	sj	2008	33	0
18	sj	2008	34	0
19	sj	2008	35	0
20	sj	2008	36	0
21	sj	2008	37	0
22	sj	2008	38	0
23	sj	2008	39	0

Once the predictive model is ready, we need to use the test data to predict the total_cases and we need to obtain the output as shown in the above file.

1.3 Problem Description

Using environmental data collected by various U.S. Federal Government agencies—from the Centers for Disease Control and Prevention to the National Oceanic and Atmospheric Administration in the U.S. Department of Commerce— We have to predict the number of dengue fever cases reported each week in San Juan, Puerto Rico and Iquitos, Peru. The goal of the competition is to develop a prediction model that would be able to anticipate the cases of dengue in every country depending on a set of climate variables mentioned above.

1.4 Significance of the problem

The WHO reports that over 2.5 billion people (that's over 40% of the world's population) are at risk for dengue (meaning, they live in areas with dengue transmission), and an estimated 50-100 million infections occur worldwide each year. This, though, is likely an underestimate, as much of dengue illness is underreported, especially since up to one half of infections are asymptomatic. Indeed, a study published in Nature earlier this year, estimated the total global burden to be 390 million, with only 96 million being apparent through clinical illness. Dengue is no joke – it causes a "bone-breaking" illness and fever that basically wipes you out.

Numerous recent studies have illuminated that global distributions of human cases of dengue and other mosquito-transmitted diseases, yet the potential distributions of key vector species have not been incorporated integrally into those mapping efforts.

Many studies have found associations between climatic conditions and dengue transmission. In this project, we aim at identifying how various climatic factors like changes in temperature, precipitation, vegetation, etc. contribute to the dengue disease.

1.5 Applicability to Data Scientists

Data Science has its applications in every area and Health care is one such wide area where it is applied. Good data-driven systems for tracking and predicting the spread of dengue disease helps the Government officials improve research initiatives and resource allocation to help fight life-threatening pandemics.

We hope our model helps in predicting the total_cases of dengue and its spread as much accurate as possible and there by solve one of the dangerous problem the world is facing today.

2 Literature Review

For better understanding of any problem and to come up with good ideas and methods, a thorough understanding of the literature is very important. It helps us in proceeding in right direction in solving the problem.

For this project, before choosing the models for prediction, we proceeded with reviewing of various scholarly articles related to the topic. Here are the inferences that we have drawn from those articles and readings:

Literature source-1:

Colon-Gonzalez FJ, Lake IR, Bentham G: Climate variability and dengue fever in warm and humid Mexico. Am J Trop Med Hyg. 2011, 84: 757-763. 10.4269/ajtmh.2011.10-0609.

Inference from the readings:

For understanding the connection between the Dengue disease spread and various climatic factors, they have used multiple linear regression models to examine the associations between changes in the climate variability and dengue incidence in the warm and humid regions of Mexico for the years 1985–2007. Their results showed that the incidence was higher during El- Niño events and in the warm and wet season. Their study demonstrated that dengue incidence was positively associated with the strength of El-Niño and the minimum temperature, especially during the cool and dry season.

Literature source-2:

Gharbi M, Quenel P, Gustave J, Cassadou S, La Ruche G, Girdary L, Marrama L: Time series analysis of dengue incidence in Guadeloupe, French West Indies: forecasting models using climate variables as predictors. BMC Infect Dis. 2011, 11: 166-10.1186/1471-2334-11-166.

Inference from the readings:

In this paper, they have applied Time series modelling approaches in assessing the impact of climate variables on dengue incidence. For example, the data is fitted a seasonal autoregressive integrated moving average (SARIMA) model of dengue incidence and climate variables including temperature, rainfall and relative humidity in French West Indies for the period 2000–2006. They found that temperature significantly improved the ability of the model to forecast dengue incidence, but this was not so for humidity and rainfall. They also found that minimum temperature at 5 weeks lag time was the best climatic variable for predicting dengue outbreaks.

Literature source-3:

Hu W, Clements A, Williams G, Tong S, Mengersen K: Spatial patterns and socioecological drivers of dengue fever transmission in Queensland, Australia. Environ Health Perspect. 2012, 120: 260-266.

Inference from the readings:

In this paper, the authors have used Bayesian spatial conditional autoregressive modelling approaches have been used to demonstrate the impact of climatic, social and ecological factors on dengue in Queensland, Australia. The authors suggested that 6% increase in locally acquired dengue was observed in association with a 1-mm increase in average monthly rainfall and a 1°C increase in average monthly maximum temperature. They also reported that overseas-acquired dengue cases were increased by 1% in association with a 1-mm increase in average monthly rainfall and a 1-unit increase in average socioeconomic index, respectively.

Literature source-4:

Climate change and threat of vector-borne diseases in India: are we prepared? Ramesh C. Dhiman, Sharmila Pahwa, G. P. S. Dhillon, Aditya P. Dash

Inference from the readings:

In this article, the authors discuss about the spread of vector-borne diseases like dengue in one of the tropical countries, India. They have used PRECIS model (driven by HadRM2) at the resolution of 50×50 Km for daily temperature and relative humidity for year 2050.

Impact of climate change on dengue also reveals increase in transmission with 2 C rise in temperature in northern India. Re-emergence of kala-azar in northern parts of India and reappearance of disease mainly in southern states of India has also been discussed. The possible need to address the threat and efforts made in India have also been highlighted. The paper concludes with a positive lead that with better preparedness threat of climate change on vector-borne diseases may be negated.

Summary and knowledge obtained:

In summary of inferences from the above articles, the quantitative models employed for evaluating the relationship between climate variables and dengue have been typically different with respect to the distributional assumptions (e.g., normal, Poisson), the nature of the relationship (linear and non-linear) and the spatial and/or temporal dynamics of the response.

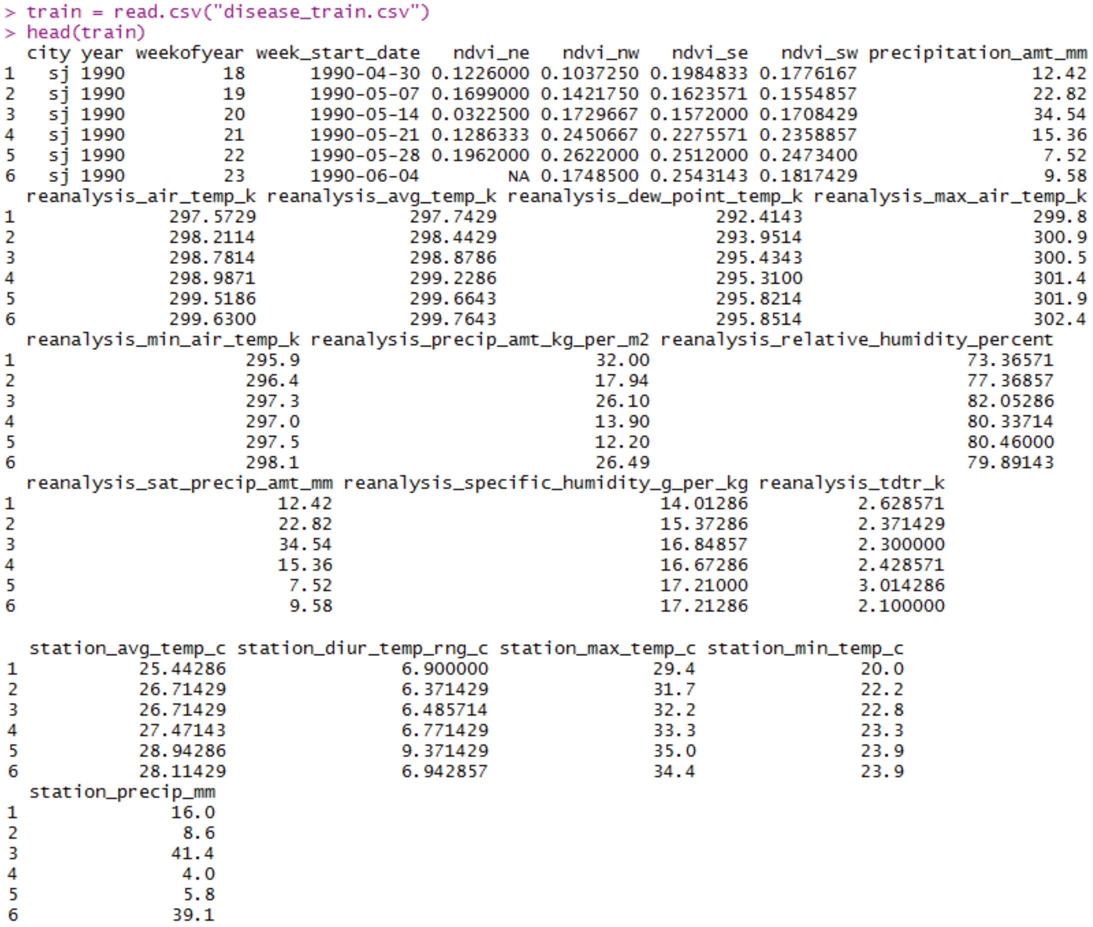
Overall, the models consistently reveal variability in the relationship between dengue and climate variables, related to country, but the methods identified an association with temperature followed by rainfall in majority of research works.

We use the knowledge obtained from these readings in choosing the models, identifying which climatic variables contribute more to the spread of dengue disease and what measures can be taken in the next steps of the project.

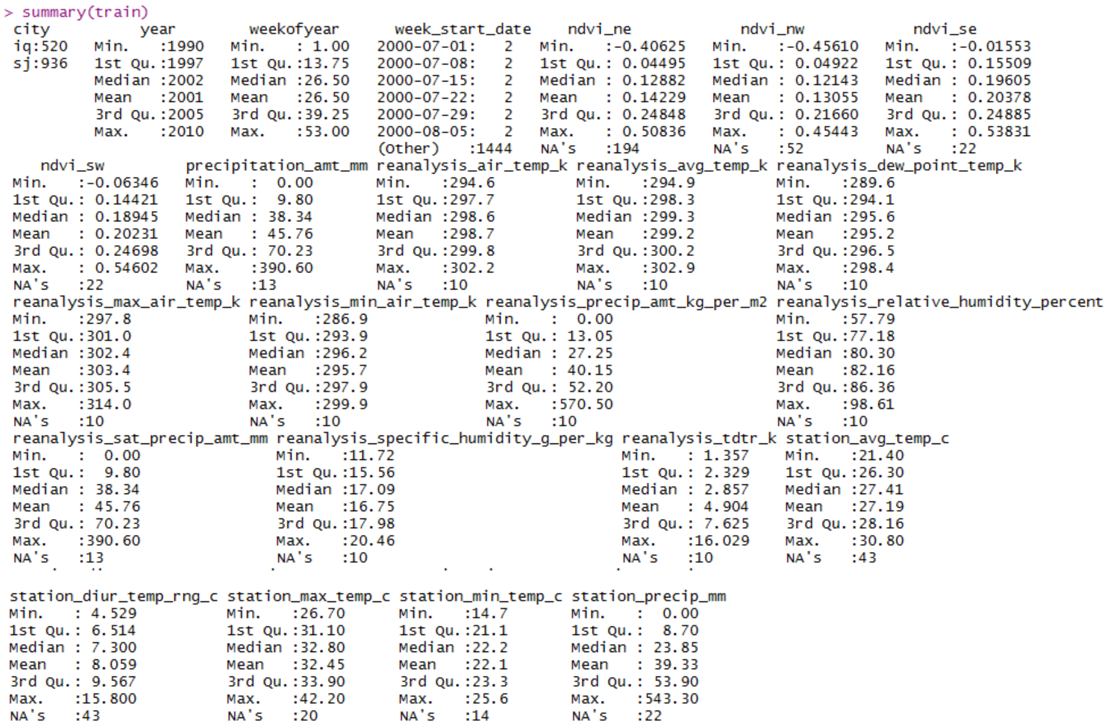
3 Data Visualization and Summary Statistics

The first step is in any prediction is load the data and analyze the shape of the data. This dataset has two cities in it: San Juan, Puerto Rico and Iquitos, Peru. Since we hypothesize that the spread of dengue may follow different patterns between the two, we need to devide the dataset, train seperate models for each city, and then join our predictions before making our final submission.

Read and summarize the original train data:



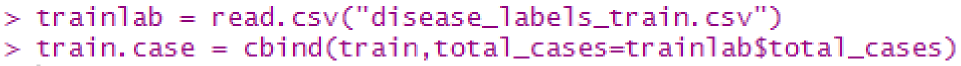
```
> train = read.csv("disease_train.csv")
> head(train)
  city year weekofyear week_start_date   ndvi_ne   ndvi_nw   ndvi_se   ndvi_sw precipitation_amt_mm
1   sj 1990         18      1990-04-30 0.1226000 0.1037250 0.1984833 0.1776167                12.42
2   sj 1990         19      1990-05-07 0.1699000 0.1421750 0.1623571 0.1554857                22.82
3   sj 1990         20      1990-05-14 0.0322500 0.1729667 0.1572000 0.1708429                34.54
4   sj 1990         21      1990-05-21 0.1286333 0.2450667 0.2275571 0.2358857                15.36
5   sj 1990         22      1990-05-28 0.1962000 0.2622000 0.2512000 0.2473400                 7.52
6   sj 1990         23      1990-06-04        NA 0.1748500 0.2543143 0.1817429                 9.58
  reanalysis_air_temp_k reanalysis_avg_temp_k reanalysis_dew_point_temp_k reanalysis_max_air_temp_k
1              297.5729              297.7429                    292.4143                     299.8
2              298.2114              298.4429                    293.9514                     300.9
3              298.7814              298.8786                    295.4343                     300.5
4              298.9871              299.2286                    295.3100                     301.4
5              299.5186              299.6643                    295.8214                     301.9
6              299.6300              299.7643                    295.8514                     302.4
  reanalysis_min_air_temp_k reanalysis_precip_amt_kg_per_m2 reanalysis_relative_humidity_percent
1                     295.9                           32.00                             73.36571
2                     296.4                           17.94                             77.36857
3                     297.3                           26.10                             82.05286
4                     297.0                           13.90                             80.33714
5                     297.5                           12.20                             80.46000
6                     298.1                           26.49                             79.89143
  reanalysis_sat_precip_amt_mm reanalysis_specific_humidity_g_per_kg reanalysis_tdtr_k
1                        12.42                              14.01286          2.628571
2                        22.82                              15.37286          2.371429
3                        34.54                              16.84857          2.300000
4                        15.36                              16.67286          2.428571
5                         7.52                              17.21000          3.014286
6                         9.58                              17.21286          2.100000

  station_avg_temp_c station_diur_temp_rng_c station_max_temp_c station_min_temp_c
1           25.44286                6.900000               29.4               20.0
2           26.71429                6.371429               31.7               22.2
3           26.71429                6.485714               32.2               22.8
4           27.47143                6.771429               33.3               23.3
5           28.94286                9.371429               35.0               23.9
6           28.11429                6.942857               34.4               23.9
  station_precip_mm
1              16.0
2               8.6
3              41.4
4               4.0
5               5.8
6              39.1
```



```
> summary(train)
 city          year        weekofyear      week_start_date     ndvi_ne              ndvi_nw              ndvi_se         
 iq:520    Min.   :1990   Min.   : 1.00   2000-07-01:   2   Min.   :-0.40625   Min.   :-0.45610   Min.   :-0.01553  
 sj:936    1st Qu.:1997   1st Qu.:13.75   2000-07-08:   2   1st Qu.: 0.04495   1st Qu.: 0.04922   1st Qu.: 0.15509  
           Median :2002   Median :26.50   2000-07-15:   2   Median : 0.12882   Median : 0.12143   Median : 0.19605  
           Mean   :2001   Mean   :26.50   2000-07-22:   2   Mean   : 0.14229   Mean   : 0.13055   Mean   : 0.20378  
           3rd Qu.:2005   3rd Qu.:39.25   2000-07-29:   2   3rd Qu.: 0.24848   3rd Qu.: 0.21660   3rd Qu.: 0.24885  
           Max.   :2010   Max.   :53.00   2000-08-05:   2   Max.   : 0.50836   Max.   : 0.45443   Max.   : 0.53831  
                                          (Other)   :1444   NA's   :194        NA's   :52         NA's   :22        
    ndvi_sw          precipitation_amt_mm reanalysis_air_temp_k reanalysis_avg_temp_k reanalysis_dew_point_temp_k
 Min.   :-0.06346   Min.   :  0.00       Min.   :294.6         Min.   :294.9         Min.   :289.6              
 1st Qu.: 0.14421   1st Qu.:  9.80       1st Qu.:297.7         1st Qu.:298.3         1st Qu.:294.1              
 Median : 0.18945   Median : 38.34       Median :298.6         Median :299.3         Median :295.6              
 Mean   : 0.20231   Mean   : 45.76       Mean   :298.7         Mean   :299.2         Mean   :295.2              
 3rd Qu.: 0.24698   3rd Qu.: 70.23       3rd Qu.:299.8         3rd Qu.:300.2         3rd Qu.:296.5              
 Max.   : 0.54602   Max.   :390.60       Max.   :302.2         Max.   :302.9         Max.   :298.4              
 NA's   :22         NA's   :13           NA's   :10            NA's   :10            NA's   :10                 
 reanalysis_max_air_temp_k reanalysis_min_air_temp_k reanalysis_precip_amt_kg_per_m2 reanalysis_relative_humidity_percent
 Min.   :297.8             Min.   :286.9             Min.   :  0.00                  Min.   :57.79                       
 1st Qu.:301.0             1st Qu.:293.9             1st Qu.: 13.05                  1st Qu.:77.18                       
 Median :302.4             Median :296.2             Median : 27.25                  Median :80.30                       
 Mean   :303.4             Mean   :295.7             Mean   : 40.15                  Mean   :82.16                       
 3rd Qu.:305.5             3rd Qu.:297.9             3rd Qu.: 52.20                  3rd Qu.:86.36                       
 Max.   :314.0             Max.   :299.9             Max.   :570.50                  Max.   :98.61                       
 NA's   :10                NA's   :10                NA's   :10                      NA's   :10                          
 reanalysis_sat_precip_amt_mm reanalysis_specific_humidity_g_per_kg reanalysis_tdtr_k station_avg_temp_c
 Min.   :  0.00               Min.   :11.72                         Min.   : 1.357    Min.   :21.40     
 1st Qu.:  9.80               1st Qu.:15.56                         1st Qu.: 2.329    1st Qu.:26.30     
 Median : 38.34               Median :17.09                         Median : 2.857    Median :27.41     
 Mean   : 45.76               Mean   :16.75                         Mean   : 4.904    Mean   :27.19     
 3rd Qu.: 70.23               3rd Qu.:17.98                         3rd Qu.: 7.625    3rd Qu.:28.16     
 Max.   :390.60               Max.   :20.46                         Max.   :16.029    Max.   :30.80     
 NA's   :13                   NA's   :10                            NA's   :10        NA's   :43        

 station_diur_temp_rng_c station_max_temp_c station_min_temp_c station_precip_mm
 Min.   : 4.529          Min.   :26.70      Min.   :14.7       Min.   :  0.00   
 1st Qu.: 6.514          1st Qu.:31.10      1st Qu.:21.1       1st Qu.:  8.70   
 Median : 7.300          Median :32.80      Median :22.2       Median : 23.85   
 Mean   : 8.059          Mean   :32.45      Mean   :22.1       Mean   : 39.33   
 3rd Qu.: 9.567          3rd Qu.:33.90      3rd Qu.:23.3       3rd Qu.: 53.90   
 Max.   :15.800          Max.   :42.20      Max.   :25.6       Max.   :543.30   
 NA's   :43              NA's   :20         NA's   :14         NA's   :22       
```

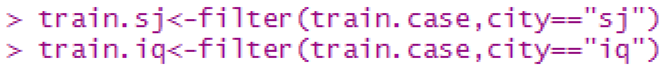
From summary of the train data we can see that this dataset has 24 variables with 1456 observations of two cities. Most of the variables are climate variables which makes sense that the transmission dynamics of dengue are related to climate variables such as temperature and precipitation because it is carried by mosquitoes.

Add the total cases of the dengue to the train data:



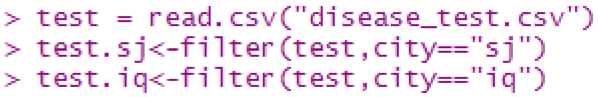
```
> trainlab = read.csv("disease_labels_train.csv")
> train.case = cbind(train,total_cases=trainlab$total_cases)
```

Separate the train data by city:



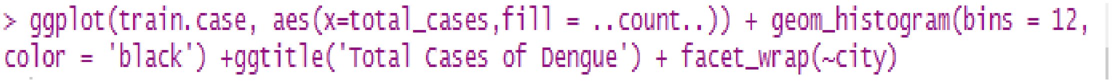
```
> train.sj<-filter(train.case,city=="sj")
> train.iq<-filter(train.case,city=="iq")
```

Separate the test data by city:

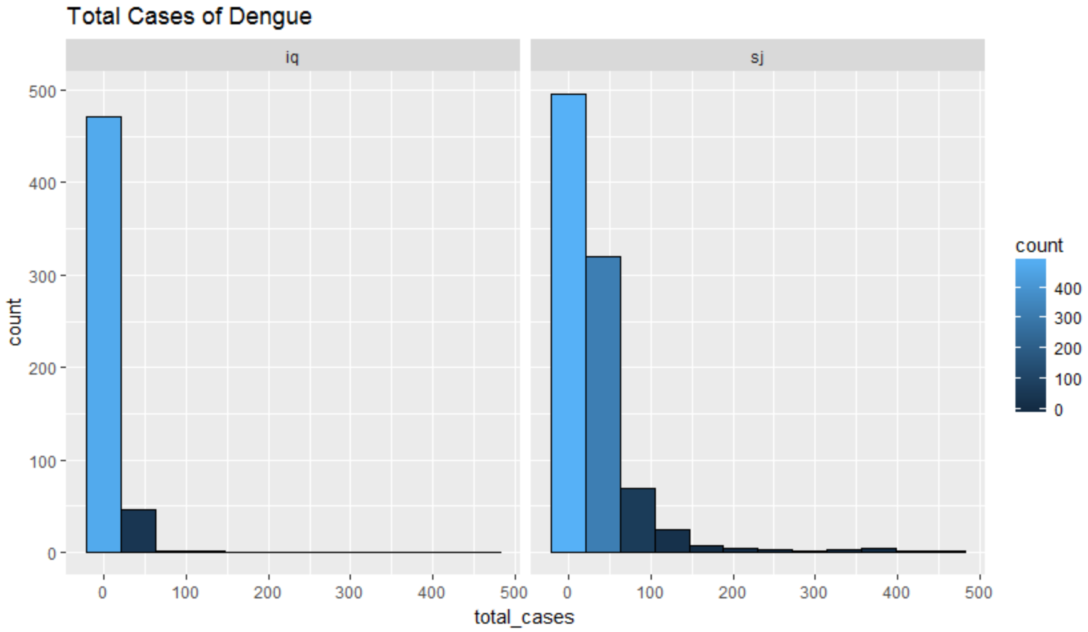


```
> test = read.csv("disease_test.csv")
> test.sj<-filter(test,city=="sj")
> test.iq<-filter(test,city=="iq")
```

Plot the total cases of dengue by city:

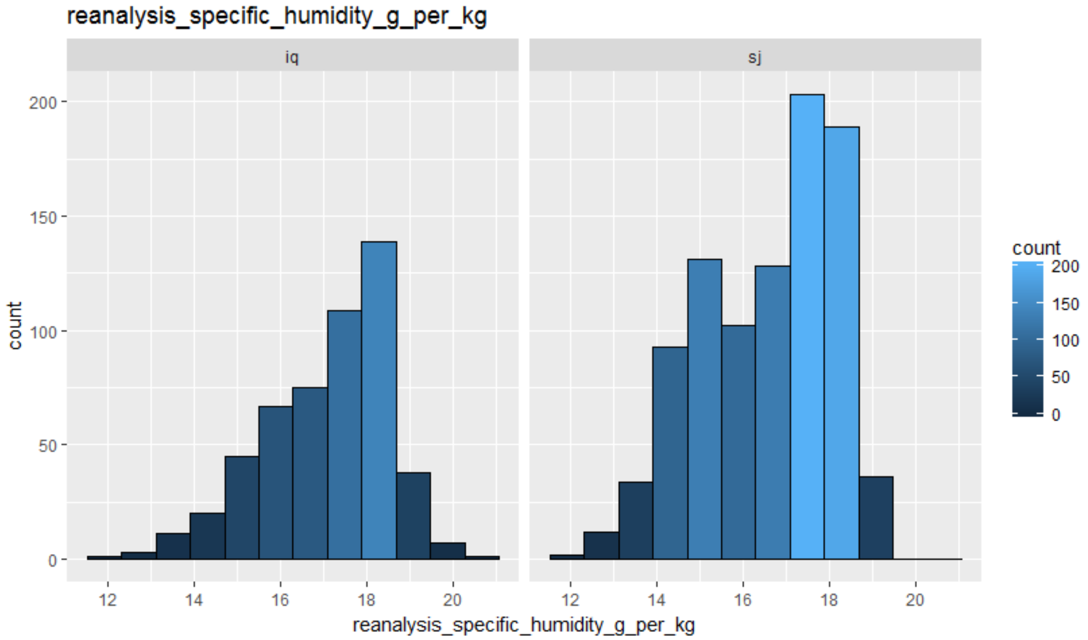


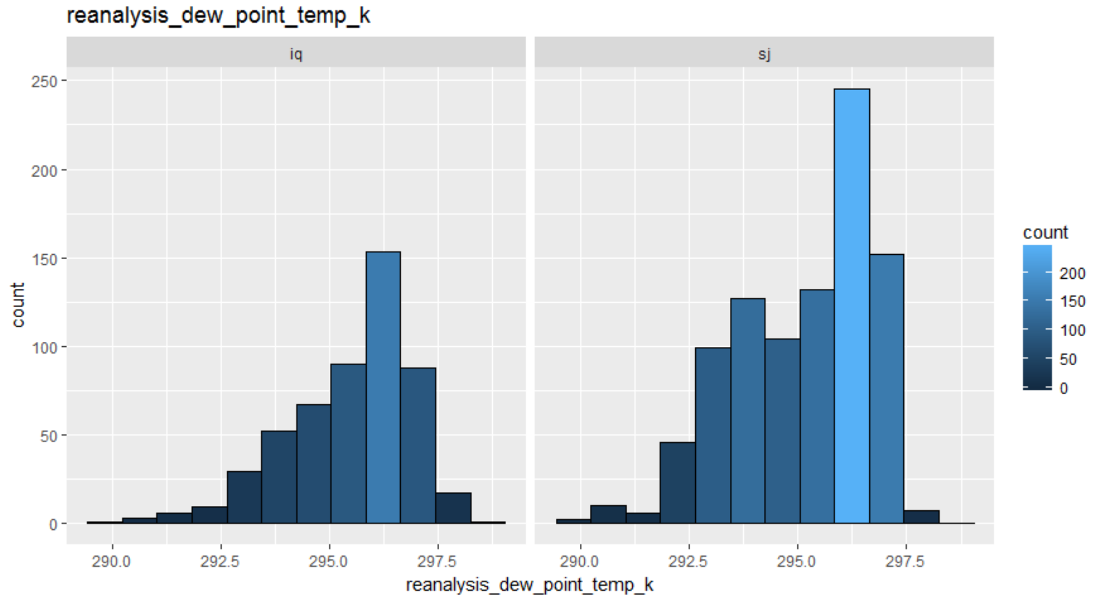
```
> ggplot(train.case, aes(x=total_cases,fill = ..count..)) + geom_histogram(bins = 12,
color = 'black') +ggtitle('Total Cases of Dengue') + facet_wrap(~city)
```

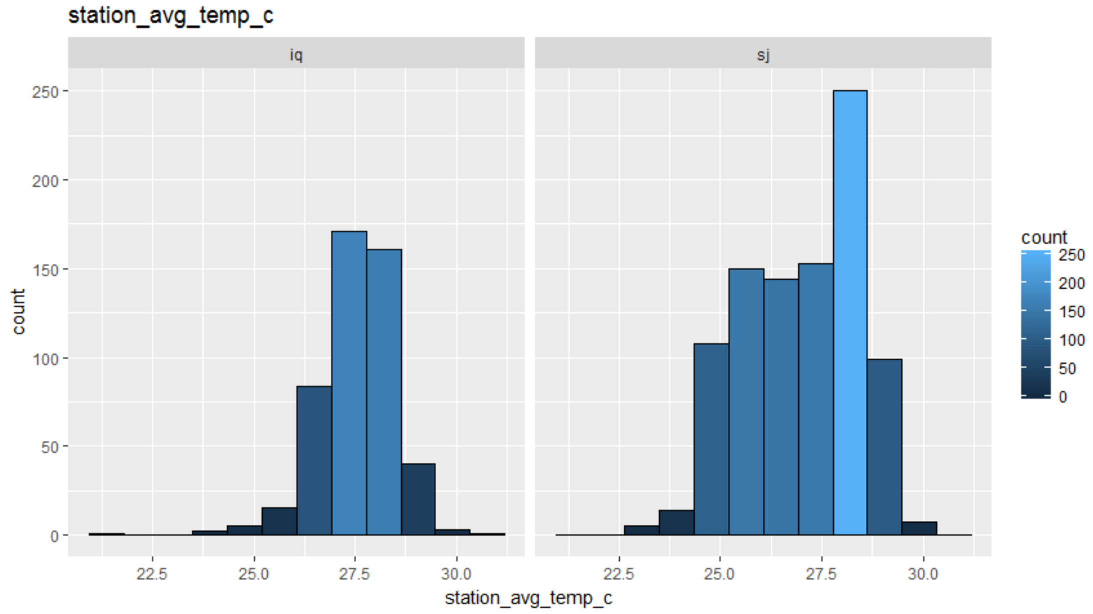


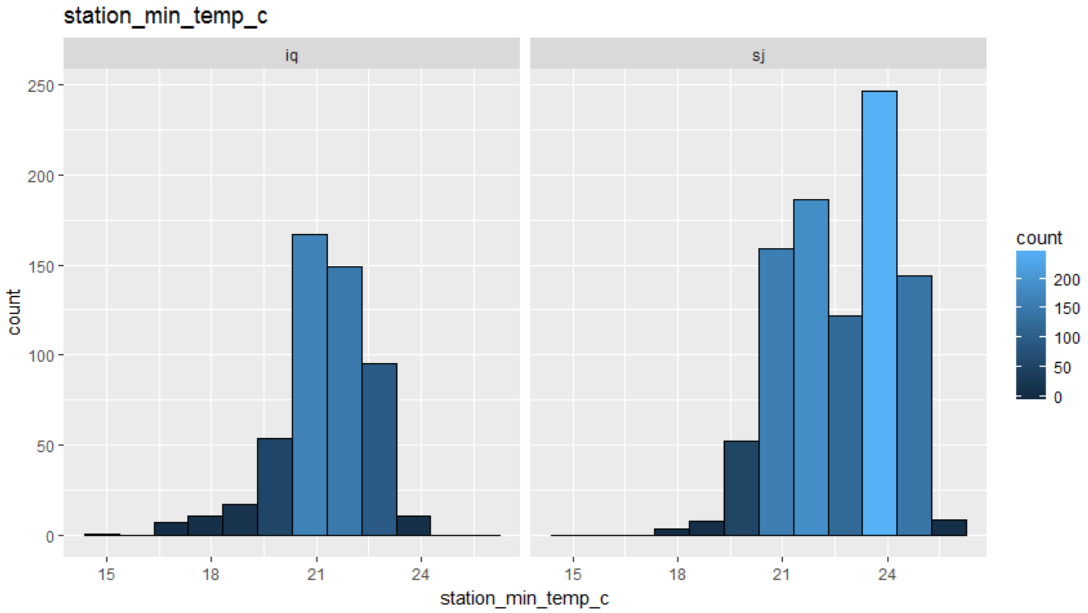
From the above plot we can clearly see that the total cases of dengue is much higher in Iquitos than that of San Juan for a certain period. And Let's see some climate variables also.

Plot the reanalysis_specific_humidity_g_per_kg , reanalysis_dew_point_temp_k, station_avg_temp_c, station_min_temp_c by city:



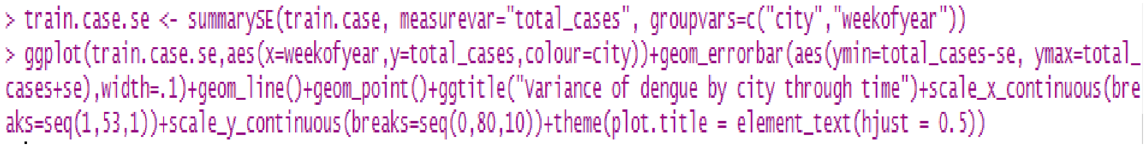

reanalysis_dew_point_temp_k
iq
sj
count
250
200
150
100
50
0
290.0
292.5
295.0
297.5
reanalysis_dew_point_temp_k
count
200
150
100
50
0


station_avg_temp_c
iq
sj
count
250
200
150
100
50
0
22.5
25.0
27.5
30.0
station_avg_temp_c
count
250
200
150
100
50
0

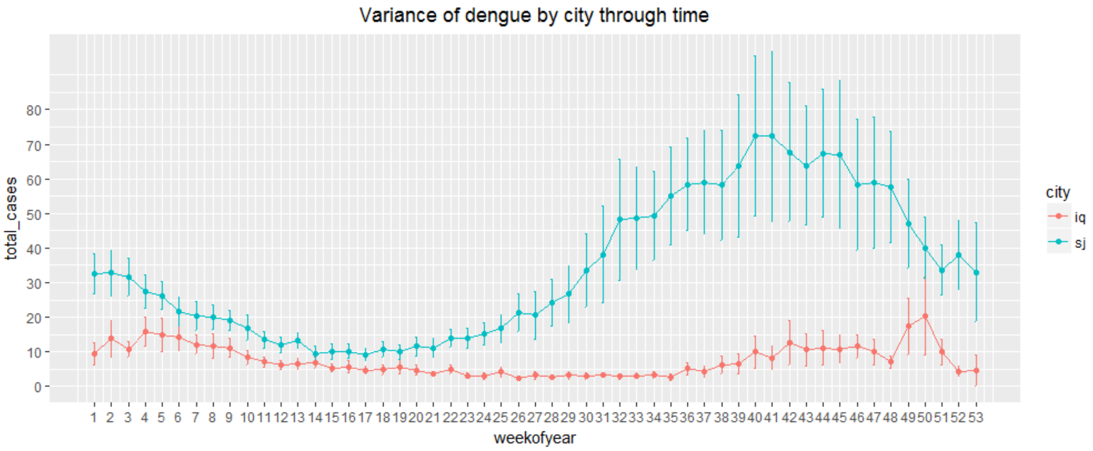


The above plots show that Iquitos is generally hotter than San Juan and the humidity is also much higher. Those climate variables might be the reasons for that total cases of dengue is more in Iquitos.

Plot the variance of dengue by city through time:



```
> train.case.se <- summarySE(train.case, measurevar="total_cases", groupvars=c("city","weekofyear"))
> ggplot(train.case.se,aes(x=weekofyear,y=total_cases,colour=city))+geom_errorbar(aes(ymin=total_cases-se, ymax=total_
cases+se),width=.1)+geom_line()+geom_point()+ggtitle("Variance of dengue by city through time")+scale_x_continuous(bre
aks=seq(1,53,1))+scale_y_continuous(breaks=seq(0,80,10))+theme(plot.title = element_text(hjust = 0.5))
```

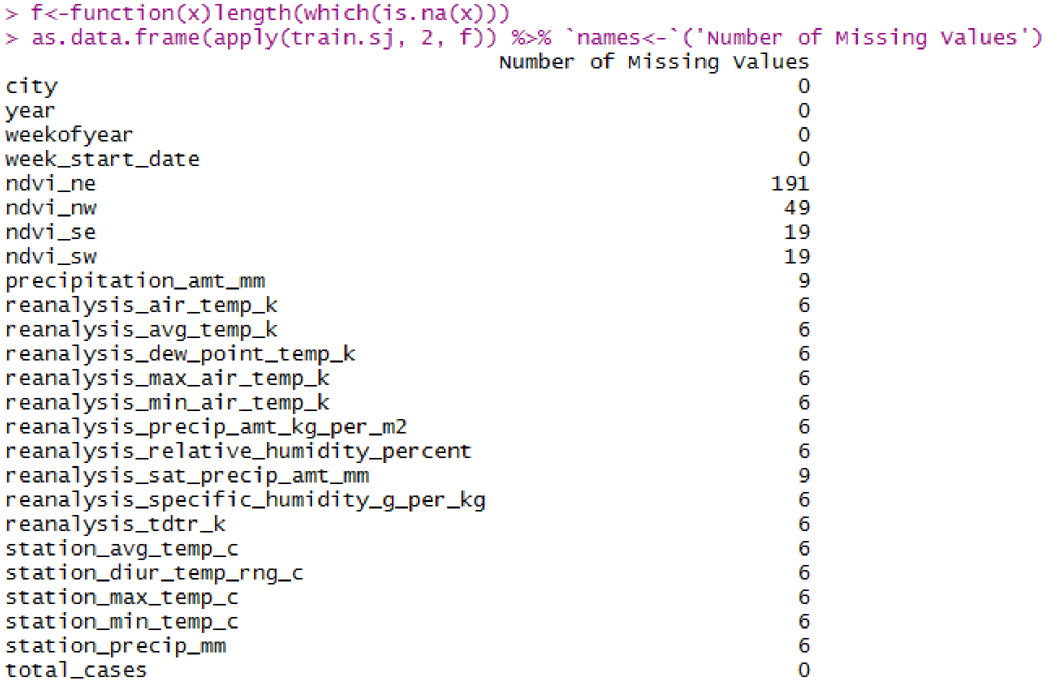


From above plot we can see that the total cases of dengue becomes a lot more active at the latter half of the year in San Juan. In Iquitos, the dengue spreads much slower with its peak at the end of the year. Those trends are of great significance in our next prediction progress.

4 Data Cleaning

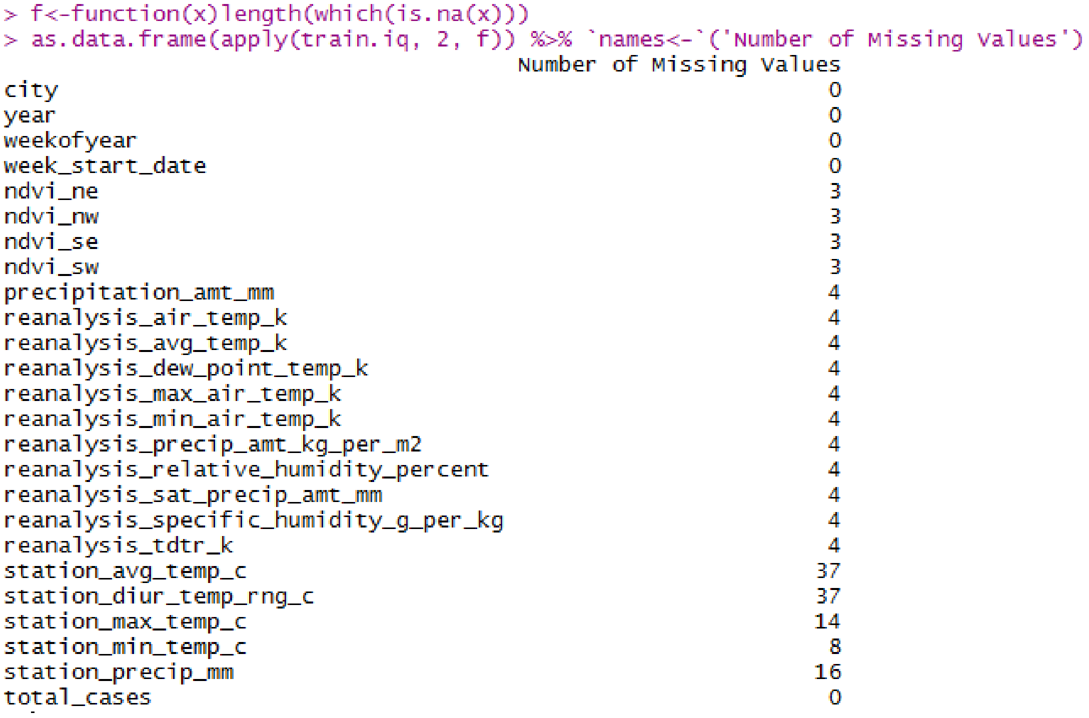
As what we see in the train data, there are many missing values in this dataset. So we should find the missing values in all subset dataframes first and then fill them.

Find number of missing values in each columns in train.sj:



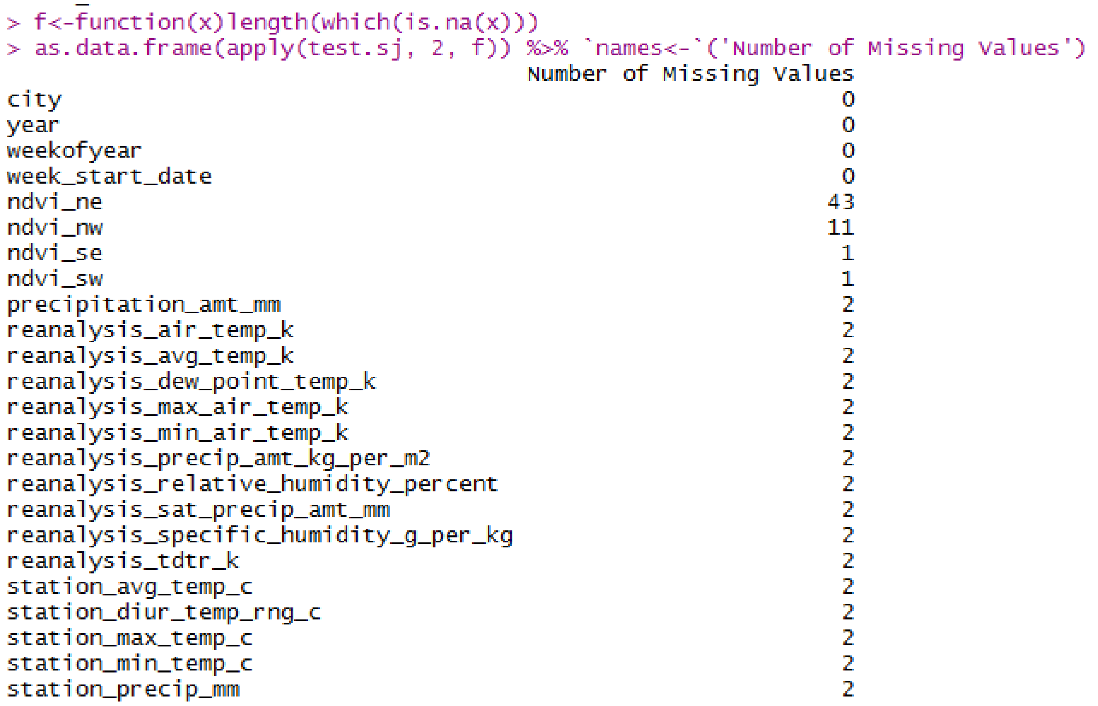
```
> f<-function(x)length(which(is.na(x)))
> as.data.frame(apply(train.sj, 2, f)) %>% `names<-`('Number of Missing Values')
                                      Number of Missing Values
city                                                         0
year                                                         0
weekofyear                                                   0
week_start_date                                              0
ndvi_ne                                                    191
ndvi_nw                                                     49
ndvi_se                                                     19
ndvi_sw                                                     19
precipitation_amt_mm                                         9
reanalysis_air_temp_k                                        6
reanalysis_avg_temp_k                                        6
reanalysis_dew_point_temp_k                                  6
reanalysis_max_air_temp_k                                    6
reanalysis_min_air_temp_k                                    6
reanalysis_precip_amt_kg_per_m2                              6
reanalysis_relative_humidity_percent                         6
reanalysis_sat_precip_amt_mm                                 9
reanalysis_specific_humidity_g_per_kg                        6
reanalysis_tdtr_k                                            6
station_avg_temp_c                                           6
station_diur_temp_rng_c                                      6
station_max_temp_c                                           6
station_min_temp_c                                           6
station_precip_mm                                            6
total_cases                                                  0
```

Find number of missing values in each columns in train.iq:



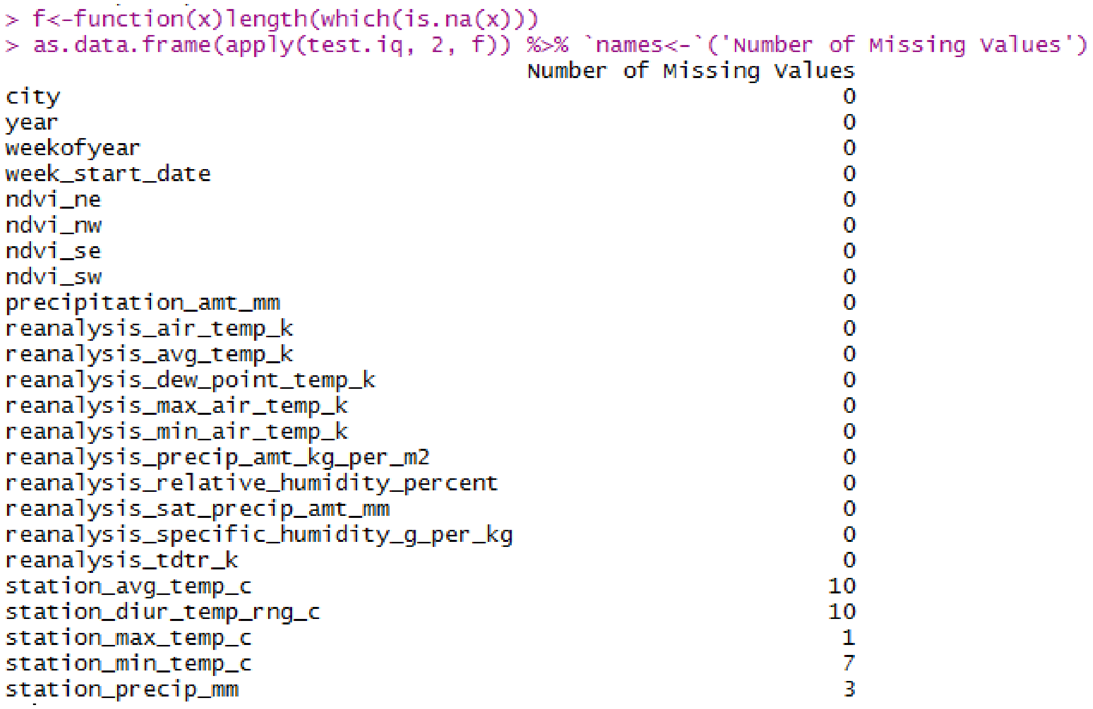
```
> f<-function(x)length(which(is.na(x)))
> as.data.frame(apply(train.iq, 2, f)) %>% `names<-`('Number of Missing Values')
                                      Number of Missing Values
city                                                         0
year                                                         0
weekofyear                                                   0
week_start_date                                              0
ndvi_ne                                                      3
ndvi_nw                                                      3
ndvi_se                                                      3
ndvi_sw                                                      3
precipitation_amt_mm                                         4
reanalysis_air_temp_k                                        4
reanalysis_avg_temp_k                                        4
reanalysis_dew_point_temp_k                                  4
reanalysis_max_air_temp_k                                    4
reanalysis_min_air_temp_k                                    4
reanalysis_precip_amt_kg_per_m2                              4
reanalysis_relative_humidity_percent                         4
reanalysis_sat_precip_amt_mm                                 4
reanalysis_specific_humidity_g_per_kg                        4
reanalysis_tdtr_k                                            4
station_avg_temp_c                                          37
station_diur_temp_rng_c                                     37
station_max_temp_c                                          14
station_min_temp_c                                           8
station_precip_mm                                           16
total_cases                                                  0
```

Find number of missing values in each columns in test.sj:



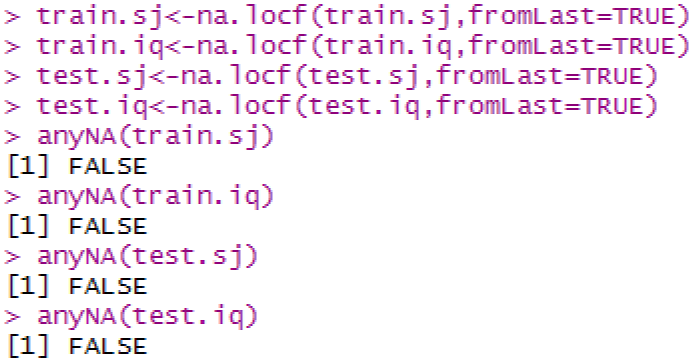
```
> f<-function(x)length(which(is.na(x)))
> as.data.frame(apply(test.sj, 2, f)) %>% `names<-`('Number of Missing Values')
                                      Number of Missing Values
city                                                         0
year                                                         0
weekofyear                                                   0
week_start_date                                              0
ndvi_ne                                                     43
ndvi_nw                                                     11
ndvi_se                                                      1
ndvi_sw                                                      1
precipitation_amt_mm                                         2
reanalysis_air_temp_k                                        2
reanalysis_avg_temp_k                                        2
reanalysis_dew_point_temp_k                                  2
reanalysis_max_air_temp_k                                    2
reanalysis_min_air_temp_k                                    2
reanalysis_precip_amt_kg_per_m2                              2
reanalysis_relative_humidity_percent                         2
reanalysis_sat_precip_amt_mm                                 2
reanalysis_specific_humidity_g_per_kg                        2
reanalysis_tdtr_k                                            2
station_avg_temp_c                                           2
station_diur_temp_rng_c                                      2
station_max_temp_c                                           2
station_min_temp_c                                           2
station_precip_mm                                            2
```

Find number of missing values in each columns in test.iq:



```
> f<-function(x)length(which(is.na(x)))
> as.data.frame(apply(test.iq, 2, f)) %>% `names<-`('Number of Missing Values')
                                      Number of Missing Values
city                                                         0
year                                                         0
weekofyear                                                   0
week_start_date                                              0
ndvi_ne                                                      0
ndvi_nw                                                      0
ndvi_se                                                      0
ndvi_sw                                                      0
precipitation_amt_mm                                         0
reanalysis_air_temp_k                                        0
reanalysis_avg_temp_k                                        0
reanalysis_dew_point_temp_k                                  0
reanalysis_max_air_temp_k                                    0
reanalysis_min_air_temp_k                                    0
reanalysis_precip_amt_kg_per_m2                              0
reanalysis_relative_humidity_percent                         0
reanalysis_sat_precip_amt_mm                                 0
reanalysis_specific_humidity_g_per_kg                        0
reanalysis_tdtr_k                                            0
station_avg_temp_c                                          10
station_diur_temp_rng_c                                     10
station_max_temp_c                                           1
station_min_temp_c                                           7
station_precip_mm                                            3
```

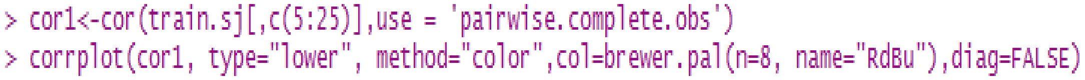
From above checking tables we can see that there many missing values in the two train data subsets. Since we can't build a model without those values, we'll take a simple approach and just fill those values with the most recent value that we saw up to that point. Filling missing values by the latest value:



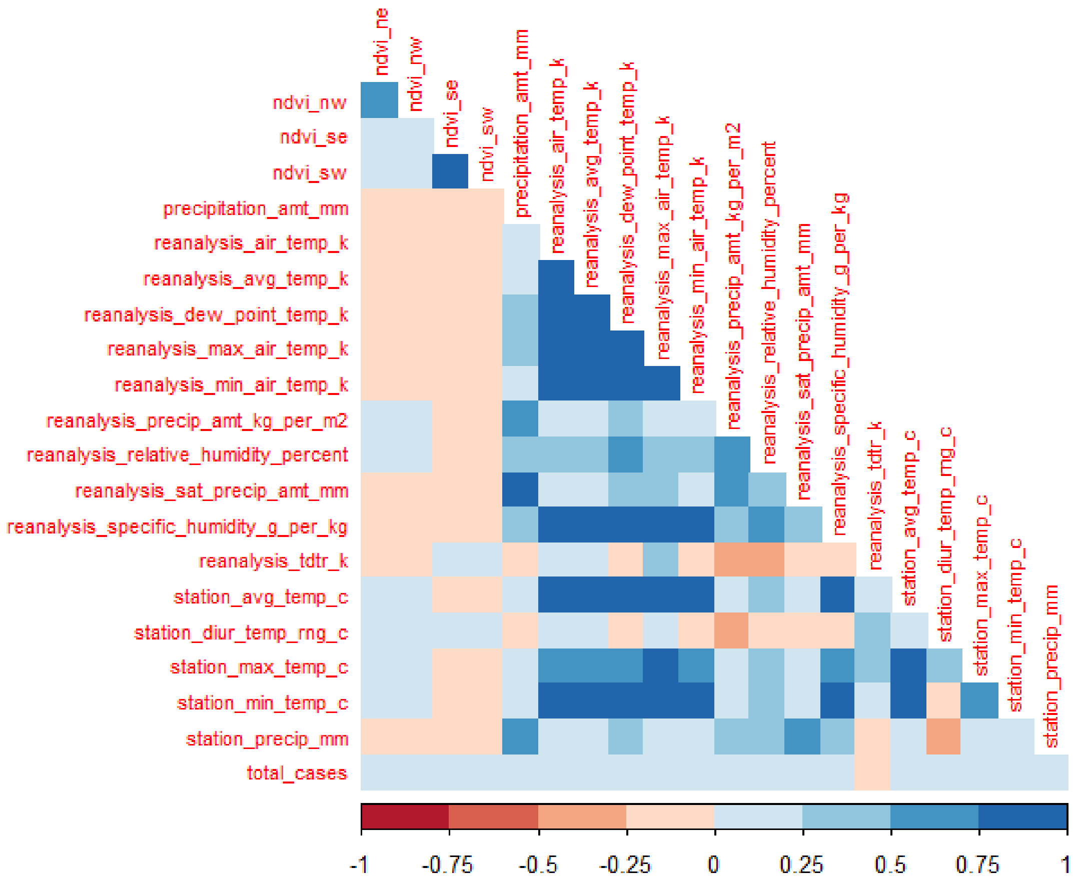
```
> train.sj<-na.locf(train.sj,fromLast=TRUE)
> train.iq<-na.locf(train.iq,fromLast=TRUE)
> test.sj<-na.locf(test.sj,fromLast=TRUE)
> test.iq<-na.locf(test.iq,fromLast=TRUE)
> anyNA(train.sj)
[1] FALSE
> anyNA(train.iq)
[1] FALSE
> anyNA(test.sj)
[1] FALSE
> anyNA(test.iq)
[1] FALSE
```

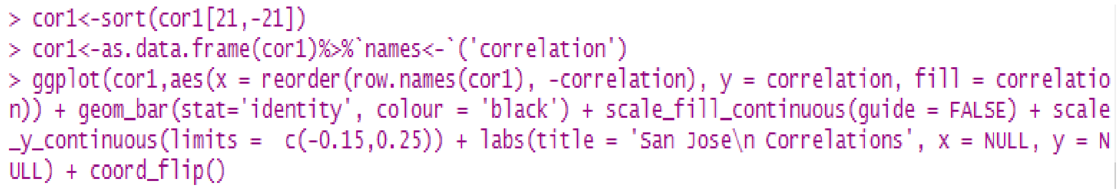
The next step in our project will be to select a subset of features to include in our prediction methods. Our primary purpose here is to get a better understanding of the relationship between the dependent viarable total cases of dengue and these independent climate variables. We will create a correlation table to show their correlations.

Plot correlation matrixs of train.sj:

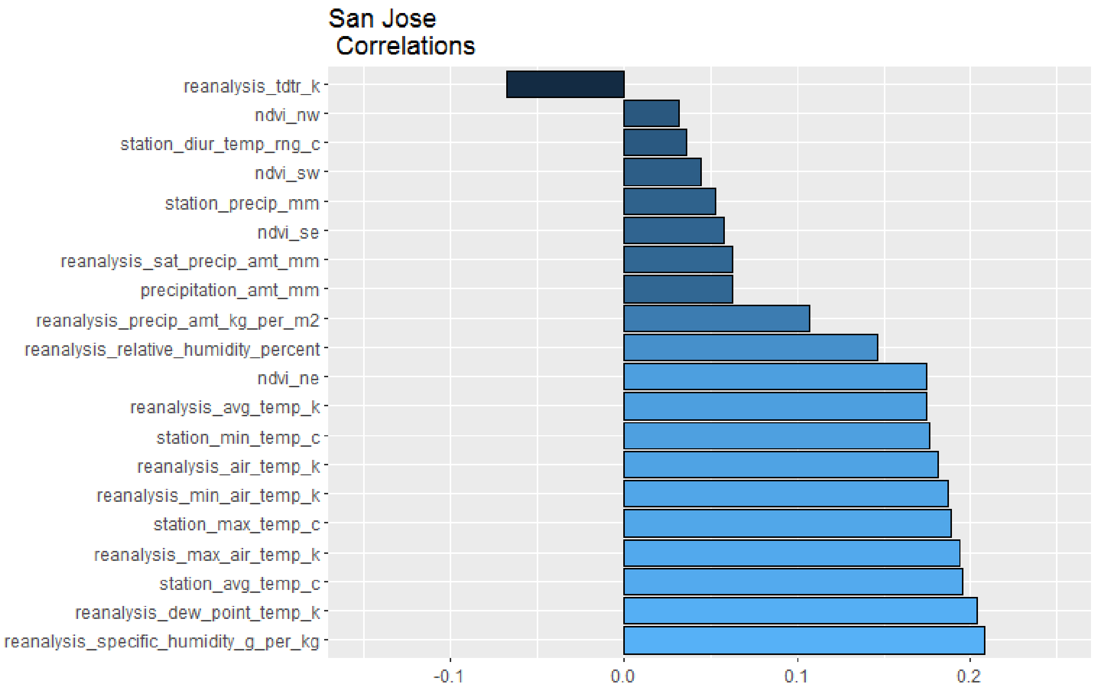


```
> cor1<-cor(train.sj[,c(5:25)],use = 'pairwise.complete.obs')
> corrplot(cor1, type="lower", method="color",col=brewer.pal(n=8, name="RdBu"),diag=FALSE)
```

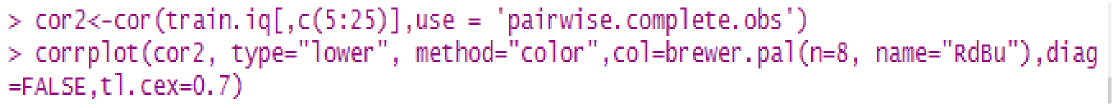




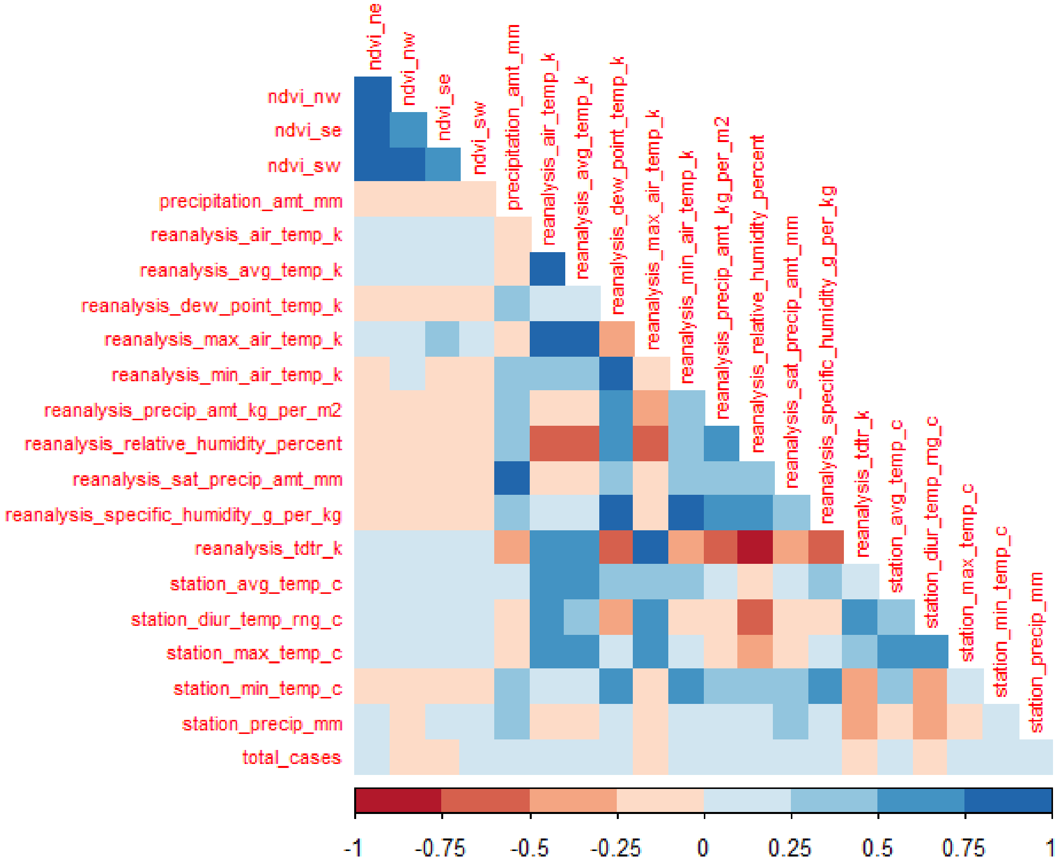
```
> cor1<-sort(cor1[21,-21])
> cor1<-as.data.frame(cor1)%>%`names<-`('correlation')
> ggplot(cor1,aes(x = reorder(row.names(cor1), -correlation), y = correlation, fill = correlatio
n)) + geom_bar(stat='identity', colour = 'black') + scale_fill_continuous(guide = FALSE) + scale
_y_continuous(limits =  c(-0.15,0.25)) + labs(title = 'San Jose\n Correlations', x = NULL, y = N
ULL) + coord_flip()
```



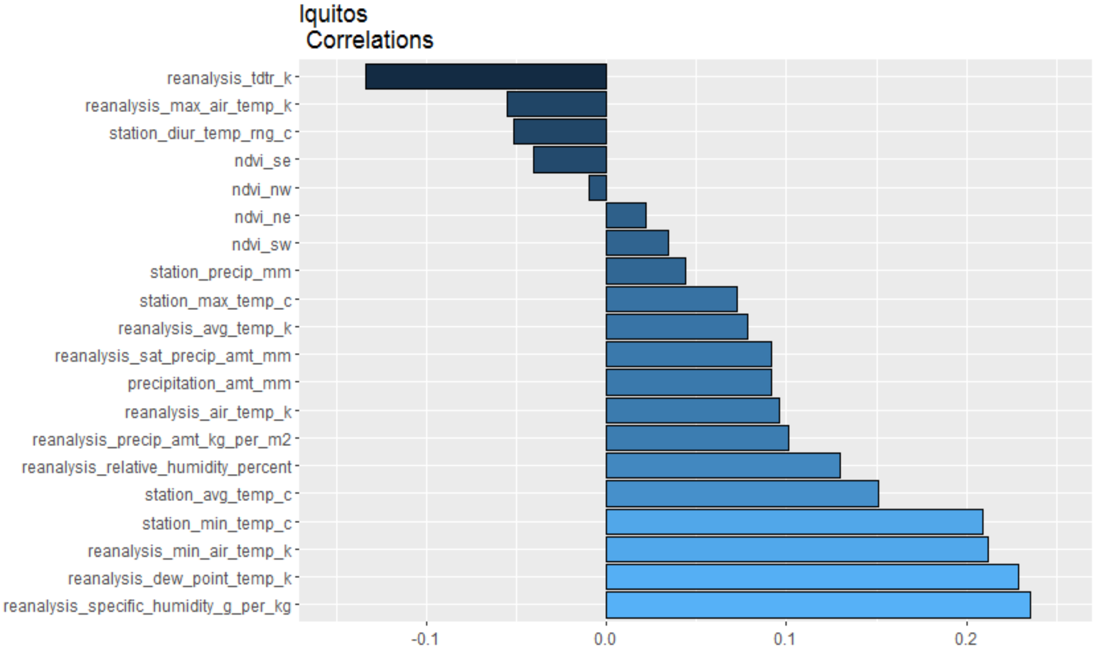
Plot correlation matrixs of train.iq:



```
> cor2<-cor(train.iq[,c(5:25)],use = 'pairwise.complete.obs')
> corrplot(cor2, type="lower", method="color",col=brewer.pal(n=8, name="RdBu"),diag
=FALSE,tl.cex=0.7)
```



```
> cor2<-sort(cor2[21,-21])
> cor2<-as.data.frame(cor2)%>%`names<-`('correlation')
> ggplot(cor2,aes(x = reorder(row.names(cor2), -correlation), y = correlation, fil
l = correlation)) + geom_bar(stat='identity', colour = 'black') + scale_fill_conti
nuous(guide = FALSE) + scale_y_continuous(limits =  c(-0.15,0.25)) + labs(title =
'Iquitos\n Correlations', x = NULL, y = NULL) + coord_flip()
```



From the correlation tables above we can see that many of the temperature data are strongly correlated, which is expected. But the total_cases variable doesn't have many obvious strong correlations. The correlation in San Jose is quite different from that in Iquitos, but it seems reanalysis_specific_humidity_g_per_kg and reanalysis_dew_point_temp_k are most strongly correlated with total cases. This makes sense that mosquitos prefer wetter climates. And we can see temperature may be another reason for denger fever due to the fact that as minimum temperatures, maximum temperatures, and average temperatures rise, the total_cases of dengue fever tend to rise as well although their correlations are not very strong. Additionally, the precipitation seems has little correlation with total cases, despite strong correlations with humidity variables.

Precisely none of these correlations are very strong. However it does not mean that we can't use these features to fit our model since at least they measure linear dependence at some point. So eventually, we will reduce some weakly correlated variables and focus on six few good variables: reanalysis_specific_humidity_g_per_kg, reanalysis_dew_point_temp_k, reanalysis_min_air_temp_k, station_min_temp_c, station_avg_temp_c, and station_max_temp_c.

5 Application of Predictive Techniques

After visualizing and understanding the data, we proceed with the application of the predictive models to the data and try to best fit the models. The data for the project is time-series in nature and we need to identify the 'total_cases' of dengue in each city for a given week of year.

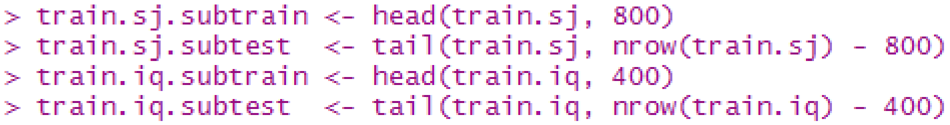
In each of the below sections, we explain how we fit the model to the data, evaluation the goodness of the fit, calculate various statistics of the data, understand the errors, Calculate AIC, BIC, how to minimize those errors, and save the final output of the model in a .csv file. We have uploaded these .csv files to the competition website and then obtained the ranks. As we applied various techniques by trying to reduce the errors, the rank improved.

5.1 Technique 1 - Negative Binomial Generalized Linear Model

5.1.1 Establish Negative Binomial Generalized Linear Model

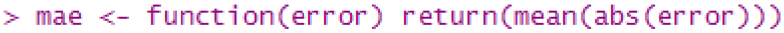
The first model is the negative binomial generalized linear model with six variables. We will use glm.nb() function in R to establish this model. Additionally, in order to find the best model we need to find the best hyper parameter alpha first. And to do this we have to split the training data up and build some new functions to find the best parameters. We'll keep around three quarters of the original data for training and use the rest to test.

Split the training data up:

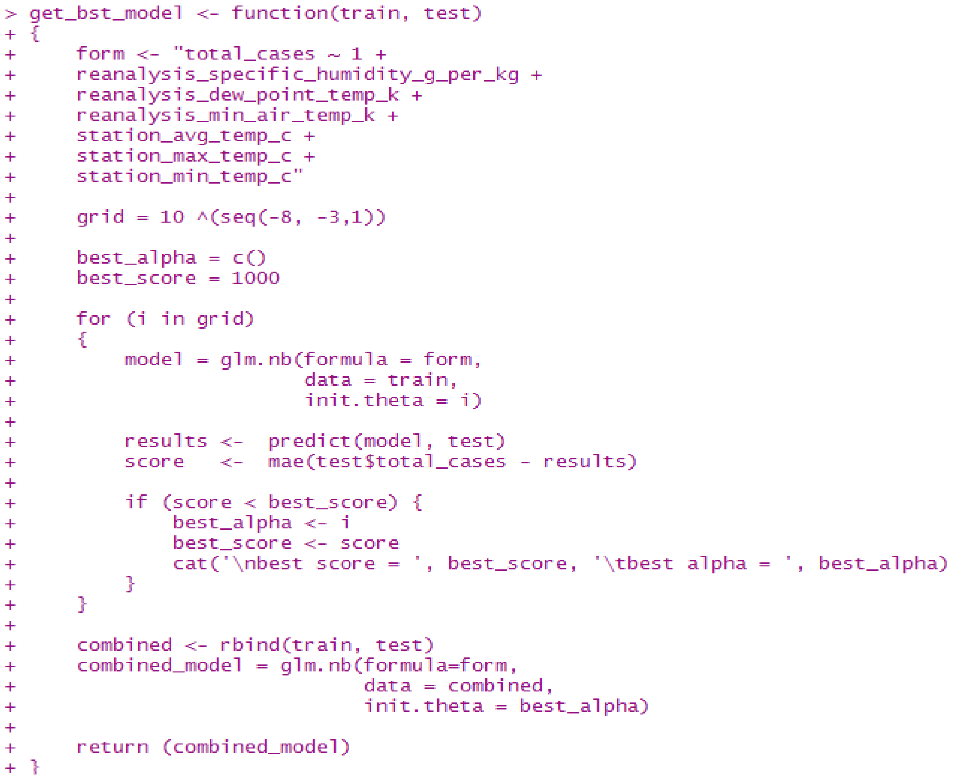


```
> train.sj.subtrain <- head(train.sj, 800)
> train.sj.subtest  <- tail(train.sj, nrow(train.sj) - 800)
> train.iq.subtrain <- head(train.iq, 400)
> train.iq.subtest  <- tail(train.iq, nrow(train.iq) - 400)
```

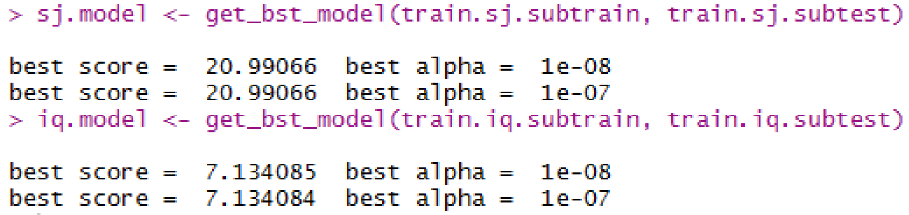
Create new functions to find the best alpha by finding the model with the minimum mean absolute error:



```
> mae <- function(error) return(mean(abs(error)))
```

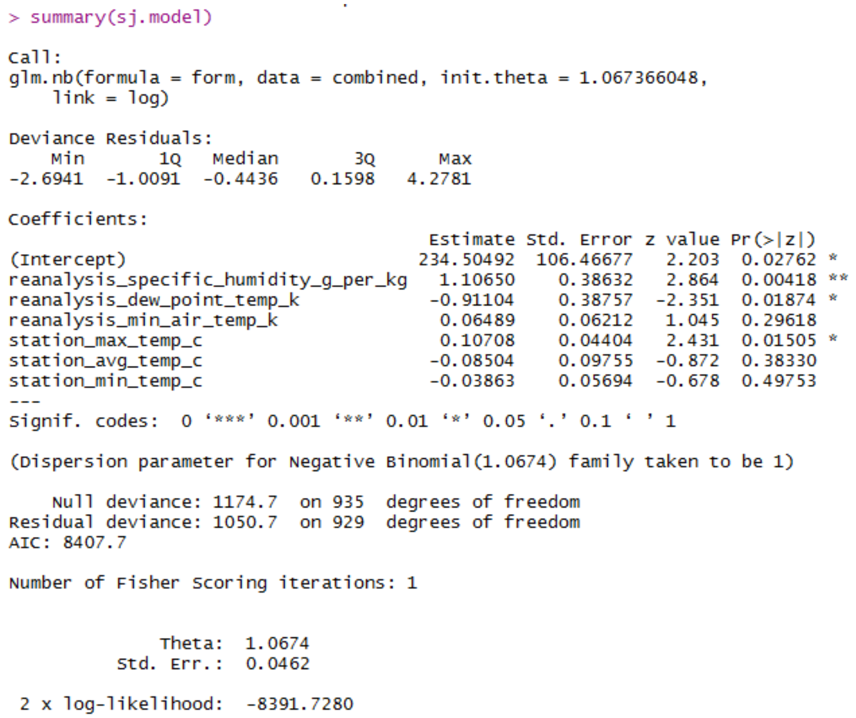


```
> get_bst_model <- function(train, test)
+ {
+     form <- "total_cases ~ 1 +
+     reanalysis_specific_humidity_g_per_kg +
+     reanalysis_dew_point_temp_k +
+     reanalysis_min_air_temp_k +
+     station_avg_temp_c +
+     station_max_temp_c +
+     station_min_temp_c"
+
+     grid = 10 ^(seq(-8, -3,1))
+
+     best_alpha = c()
+     best_score = 1000
+
+     for (i in grid)
+     {
+         model = glm.nb(formula = form,
+                        data = train,
+                        init.theta = i)
+
+         results <-  predict(model, test)
+         score   <-  mae(test$total_cases - results)
+
+         if (score < best_score) {
+             best_alpha <- i
+             best_score <- score
+             cat('\nbest score = ', best_score, '\tbest alpha = ', best_alpha)
+         }
+     }
+
+     combined <- rbind(train, test)
+     combined_model = glm.nb(formula=form,
+                             data = combined,
+                             init.theta = best_alpha)
+
+     return (combined_model)
+ }
```



```
> sj.model <- get_bst_model(train.sj.subtrain, train.sj.subtest)

best score =  20.99066  best alpha =  1e-08
best score =  20.99066  best alpha =  1e-07
> iq.model <- get_bst_model(train.iq.subtrain, train.iq.subtest)

best score =  7.134085  best alpha =  1e-08
best score =  7.134084  best alpha =  1e-07
```

Summarize the models:

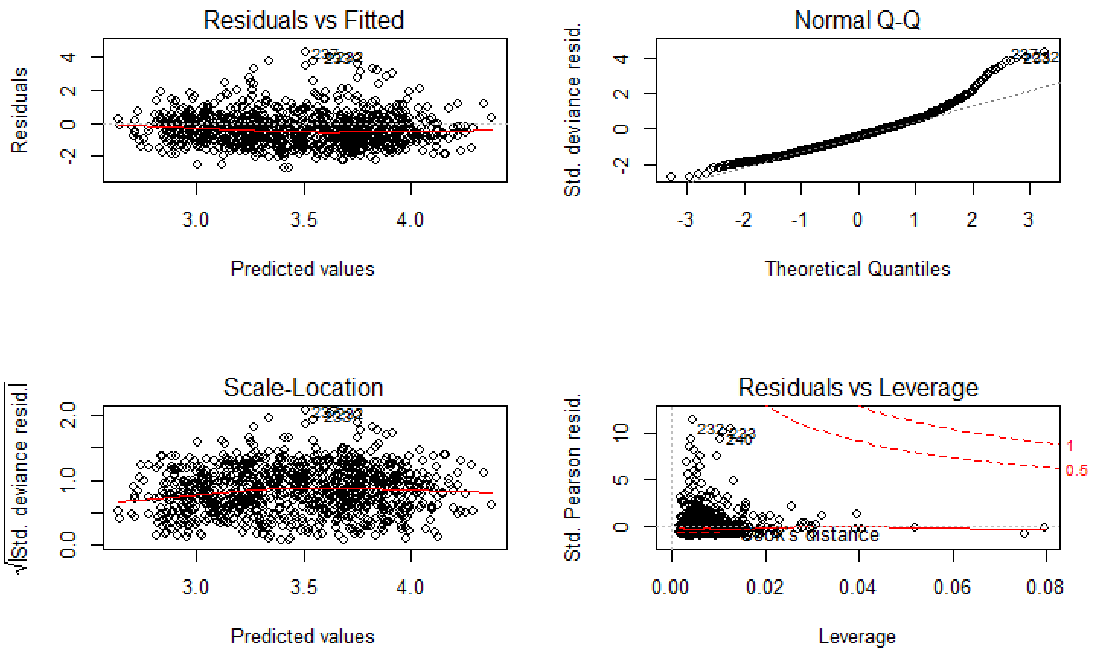


```
> summary(sj.model)

Call:
glm.nb(formula = form, data = combined, init.theta = 1.067366048,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6941  -1.0091  -0.4436   0.1598   4.2781

Coefficients:
                                       Estimate Std. Error z value Pr(>|z|)
(Intercept)                           234.50492  106.46677   2.203  0.02762 *
reanalysis_specific_humidity_g_per_kg   1.10650    0.38632   2.864  0.00418 **
reanalysis_dew_point_temp_k            -0.91104    0.38757  -2.351  0.01874 *
reanalysis_min_air_temp_k               0.06489    0.06212   1.045  0.29618
station_max_temp_c                      0.10708    0.04404   2.431  0.01505 *
station_avg_temp_c                     -0.08504    0.09755  -0.872  0.38330
station_min_temp_c                     -0.03863    0.05694  -0.678  0.49753
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(1.0674) family taken to be 1)

    Null deviance: 1174.7  on 935  degrees of freedom
Residual deviance: 1050.7  on 929  degrees of freedom
AIC: 8407.7

Number of Fisher Scoring iterations: 1


              Theta:  1.0674
          Std. Err.:  0.0462

 2 x log-likelihood:  -8391.7280
```

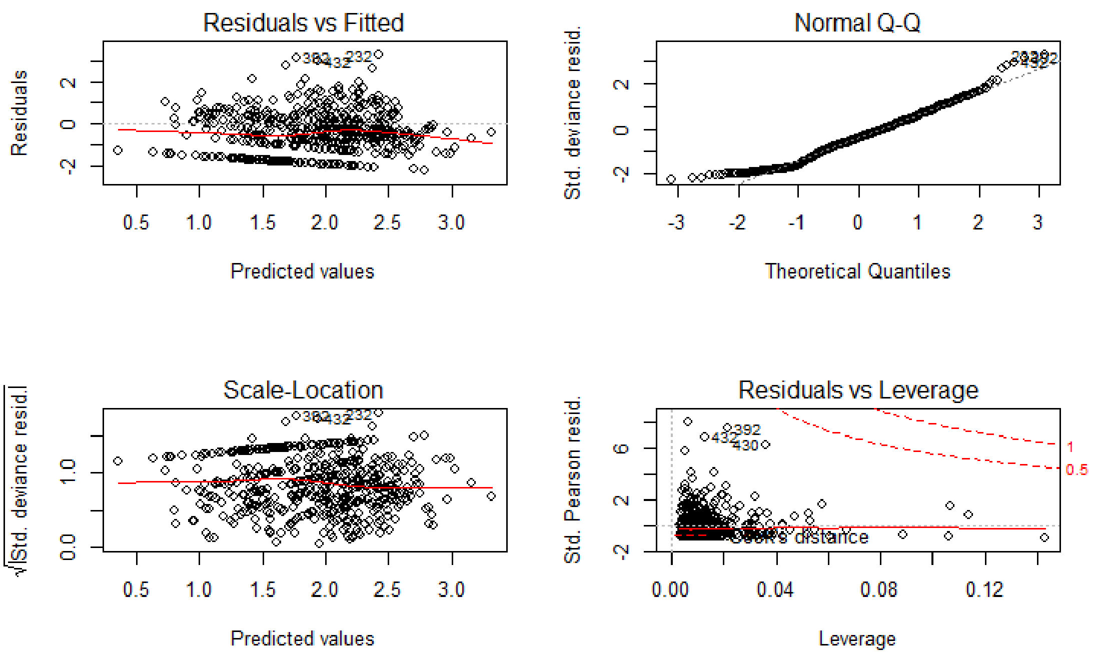
```
> summary(iq.model)

Call:
glm.nb(formula = form, data = combined, init.theta = 0.7952871889,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2051  -1.0884  -0.4291   0.2775   3.2549

Coefficients:
                                       Estimate Std. Error z value Pr(>|z|)
(Intercept)                           397.37715  165.12136   2.407  0.01610 *
reanalysis_specific_humidity_g_per_kg   1.62072    0.57667   2.810  0.00495 **
reanalysis_dew_point_temp_k            -1.48292    0.58791  -2.522  0.01166 *
reanalysis_min_air_temp_k               0.03336    0.05133   0.650  0.51577
station_max_temp_c                      0.02695    0.05421   0.497  0.61907
station_avg_temp_c                      0.04306    0.08734   0.493  0.62201
station_min_temp_c                      0.14920    0.05825   2.562  0.01042 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(0.7953) family taken to be 1)

    Null deviance: 665.57  on 519  degrees of freedom
Residual deviance: 595.22  on 513  degrees of freedom
AIC: 3130.4

Number of Fisher Scoring iterations: 1


              Theta:  0.7953
          Std. Err.:  0.0574

 2 x log-likelihood:  -3114.4460
```

Plot the models:

```
> par(mfrow=c(2,2))
> plot(sj.model)
```



```
> plot(iq.model)
```

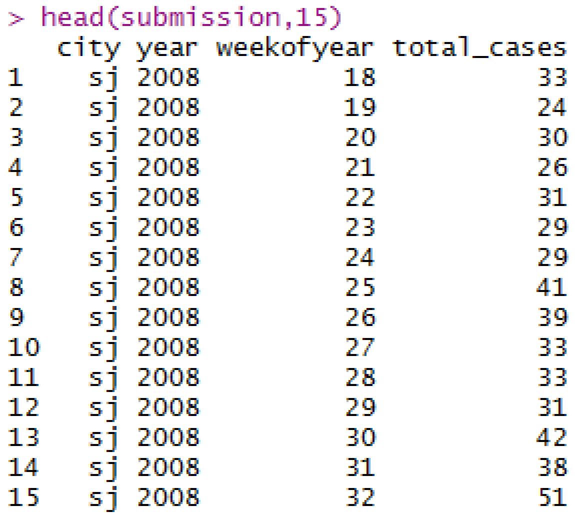


5.1.2 Predict the test data and submit the outcome

Predict the test data and submit the outcome:

```
> sj.predict <- predict(sj.model,test.sj,type = "response")
> iq.predict <- predict(iq.model,test.iq,type = "response")
> sj.prediction <- data.frame('total_cases' = sj.predict)
> iq.prediction <- data.frame('total_cases' = iq.predict)
> submission<-read.csv("submission_format.csv")
> submission$total_cases<-as.numeric(c(sj.prediction$total_cases,iq.prediction$total_cases))
> submission$total_cases<-round(submission$total_cases,0)
> write.csv(submission,file="Negative Binomial Model.csv",row.names=F)
```

Sample test prediction data:

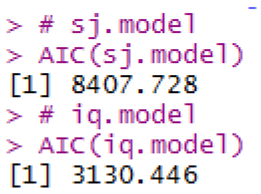


```
> head(submission,15)
   city year weekofyear total_cases
1    sj 2008         18          33
2    sj 2008         19          24
3    sj 2008         20          30
4    sj 2008         21          26
5    sj 2008         22          31
6    sj 2008         23          29
7    sj 2008         24          29
8    sj 2008         25          41
9    sj 2008         26          39
10   sj 2008         27          33
11   sj 2008         28          33
12   sj 2008         29          31
13   sj 2008         30          42
14   sj 2008         31          38
15   sj 2008         32          51
```

5.1.3 Assessment of the Negative Binomial Generalized Linear Model

i) AIC (Akaike information criterion)

AIC offers an estimate of the relative information lost when a given model is used to represent the process that generated the data. In doing so, it deals with the trade-off between the goodness of fit of the model and the complexity of the model. So, lesser the AIC value better is the model.

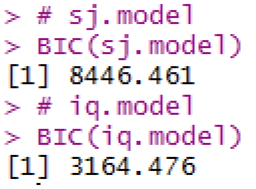


```
> # sj.model
> AIC(sj.model)
[1] 8407.728
> # iq.model
> AIC(iq.model)
[1] 3130.446
```

From above table we can see that the same model fits the data in Iquitos much better than those in San Juan. Additionally, both of the little big AIC values means that there may be some relative information lost in our models and our models can be improved further.

ii) BIC (Bayesian information criterion)

BIC is a criterion for model selection among a finite set of models; the model with the lowest BIC is preferred. It is based, in part, on the likelihood function and it is closely related to the Akaike information criterion (AIC).

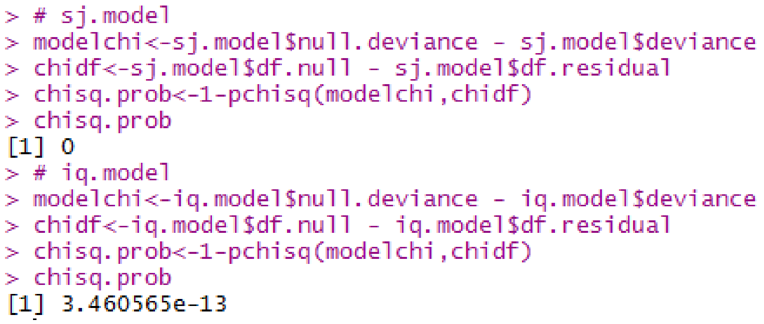


```
> # sj.model
> BIC(sj.model)
[1] 8446.461
> # iq.model
> BIC(iq.model)
[1] 3164.476
```

The outcomes of BIC above show the same meaning as those of AIC. This means the same model fits the data in Iquitos much better than those in San Juan. Maybe we should try to use different variables in the model to fit different city's data to improve our models.

iii) Chi-squared distribution

We can also use the residual deviance to test whether the null hypothesis is true (i.e. Logistic regression model provides an adequate fit for the data). This is possible because the deviance is given by the chi-squared value at certain degree of freedom. To test for significance, we can find out associated p-values using the below formula in R.

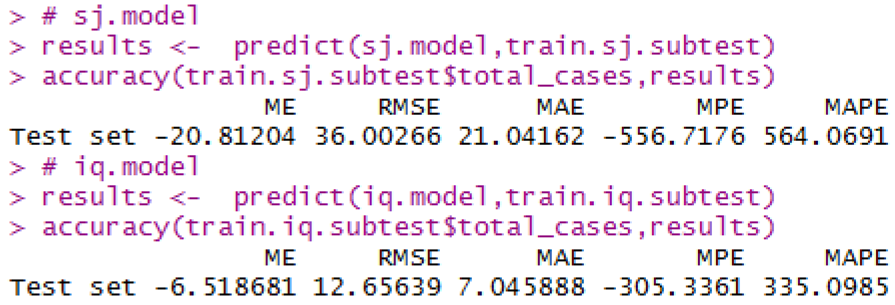


```
> # sj.model
> modelchi<-sj.model$null.deviance - sj.model$deviance
> chidf<-sj.model$df.null - sj.model$df.residual
> chisq.prob<-1-pchisq(modelchi,chidf)
> chisq.prob
[1] 0
> # iq.model
> modelchi<-iq.model$null.deviance - iq.model$deviance
> chidf<-iq.model$df.null - iq.model$df.residual
> chisq.prob<-1-pchisq(modelchi,chidf)
> chisq.prob
[1] 3.460565e-13
```

Using the above values of residual deviance and df, we get a p-value of approximately zero for both the models showing that there is a significant lack of evidence to support the null hypothesis.

iv) Measurement of Forecasting Error

There are many types of measurements of the forecasting error, including Mean Error (ME), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE). We can use these measurements to test whether the model forecasts well. In R we have accuracy ( ) function to do the test.



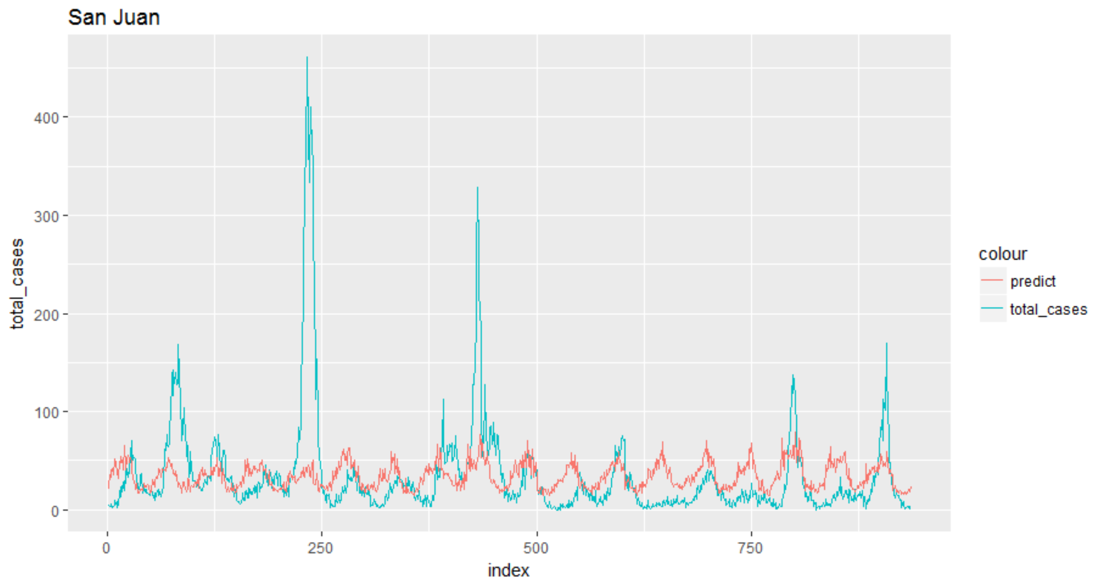
```
> # sj.model
> results <-  predict(sj.model,train.sj.subtest)
> accuracy(train.sj.subtest$total_cases,results)
                ME     RMSE      MAE       MPE     MAPE
Test set -20.81204 36.00266 21.04162 -556.7176 564.0691
> # iq.model
> results <-  predict(iq.model,train.iq.subtest)
> accuracy(train.iq.subtest$total_cases,results)
                ME     RMSE      MAE       MPE     MAPE
Test set -6.518681 12.65639 7.045888 -305.3361 335.0985
```

From above test table we can clearly see that iq.model did a much better prediction performance than the sj.model. So if we want to improve our models, we should try to decrease these errors. The Mean Absolute Error (MAE) is the evaluation metric of this project and obviously the sj.model didn't perform well on the prediction for San Juan city. So we should further find other methods to reduce the MAE in the San Juan part.

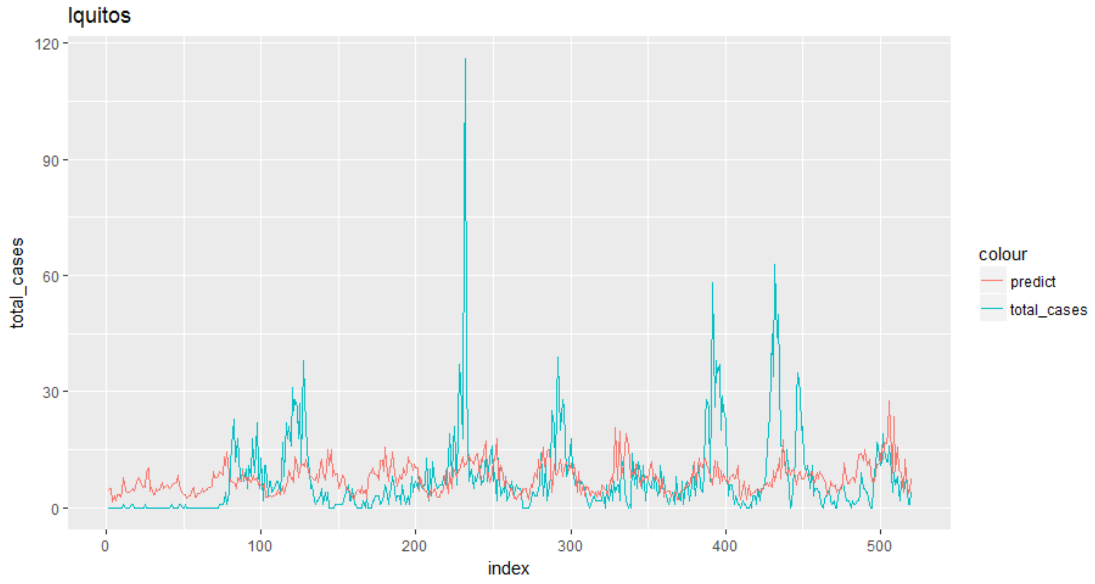
v) Use plot to test the prediction performance of each model

From above tests we have known that we can improve our model. But we don't know where our model is going wrong. However, we can have some hints about where investments will improve the model performance by showing a plot of the predicted values and actual values.

```
> # plot San Juan part
> train.sj$predict<-predict(sj.model,train.sj, type = 'response')
> train.sj<-mutate(train.sj,index = as.numeric(row.names(train.sj)))
> ggplot(train.sj,aes(x = index)) + ggtitle("San Juan") + geom_line(aes(y =
total_cases, colour = "total_cases")) + geom_line(aes(y = predict, colour =
"predict"))
```



```
> # plot Iquitos part
> train.iq$predict<-predict(iq.model,train.iq, type = 'response')
> train.iq<-mutate(train.iq,index = as.numeric(row.names(train.iq)))
> ggplot(train.iq,aes(x = index)) + ggtitle("Iquitos") + geom_line(aes(y =
total_cases, colour = "total_cases")) + geom_line(aes(y = predict, colour =
 "predict"))
```

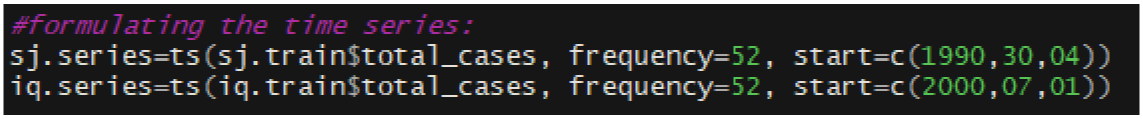


From above plots we can see that our models in red indeed track the seasonality of Dengue cases. However, the timing of the seasonality of our predictions has a mismatch with the actual results. One potential reason for this is that our features don't look far enough into the past. Because dengue is mosquito born, and the mosquito lifecycle depends on water, we need to take both the life of a mosquito and the time between infection and symptoms into account when modeling dengue. This is a critical avenue to explore when improving this model.

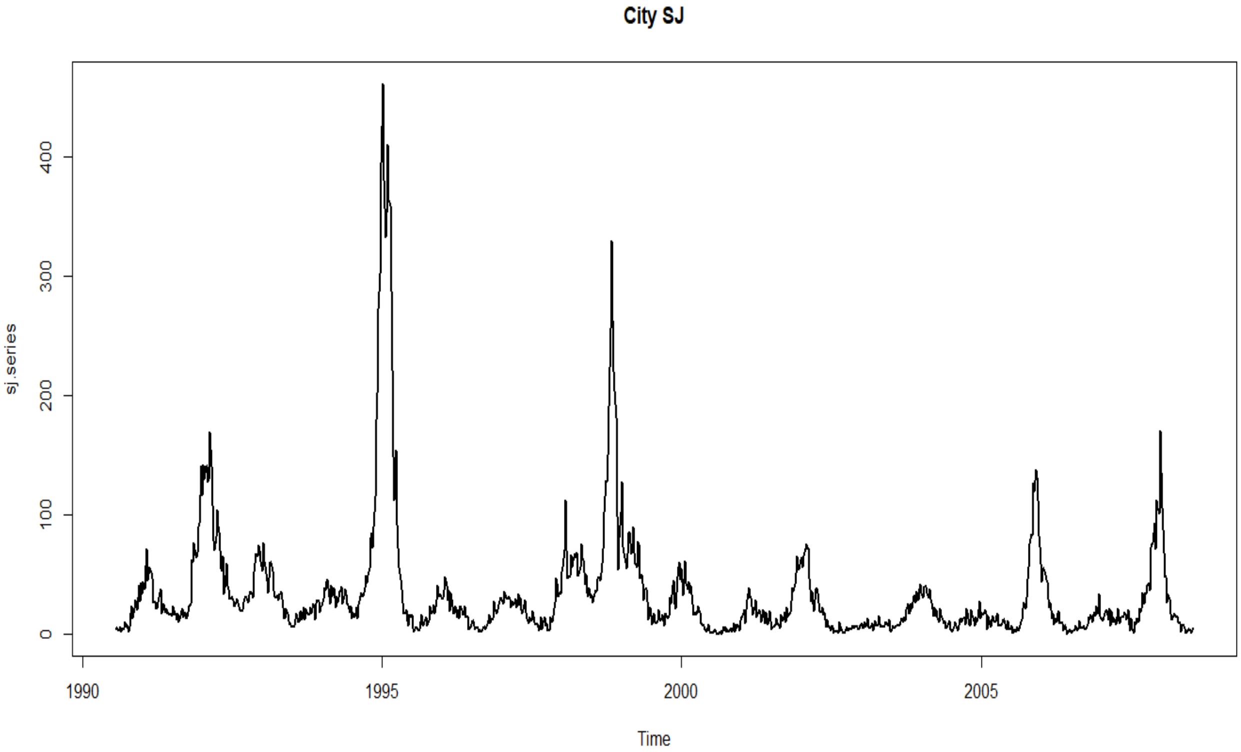
The other important error is that our predictions are relatively consistent–we miss the spikes that are large outbreaks. One reason is that we don't take into account the contagiousness of dengue. A possible way to account for this is to build a model that progressively predicts a new value while taking into account the previous prediction.

5.2 Techique 2 – Arima Model

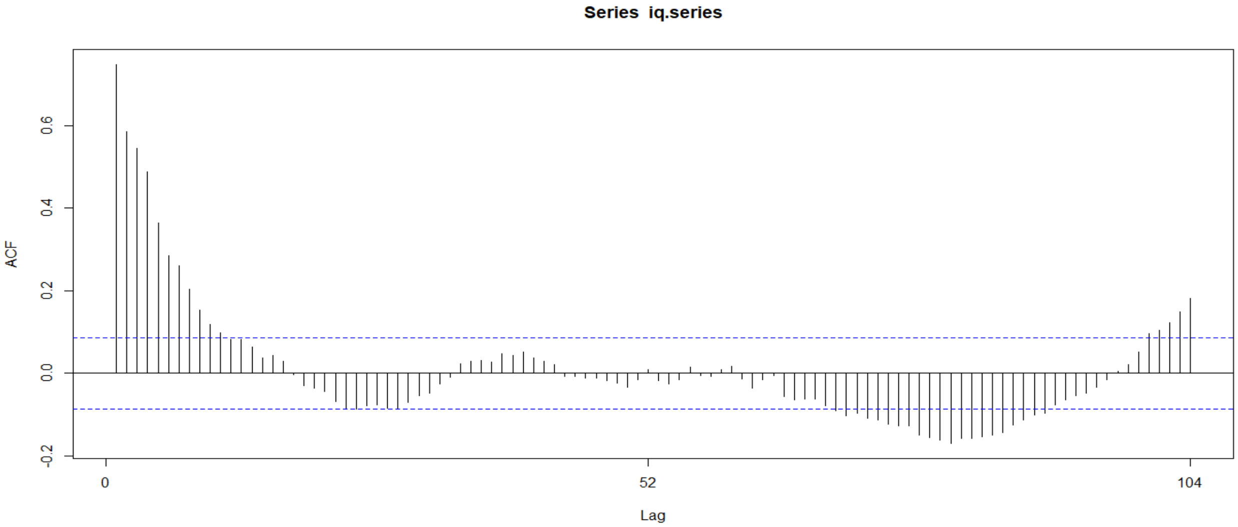
The second techique we will be using is the Arima model. Arima model is applied to the time series with no seasonality and trend. Xreg Arima and seasonal arima are the two models that applied to this time series.



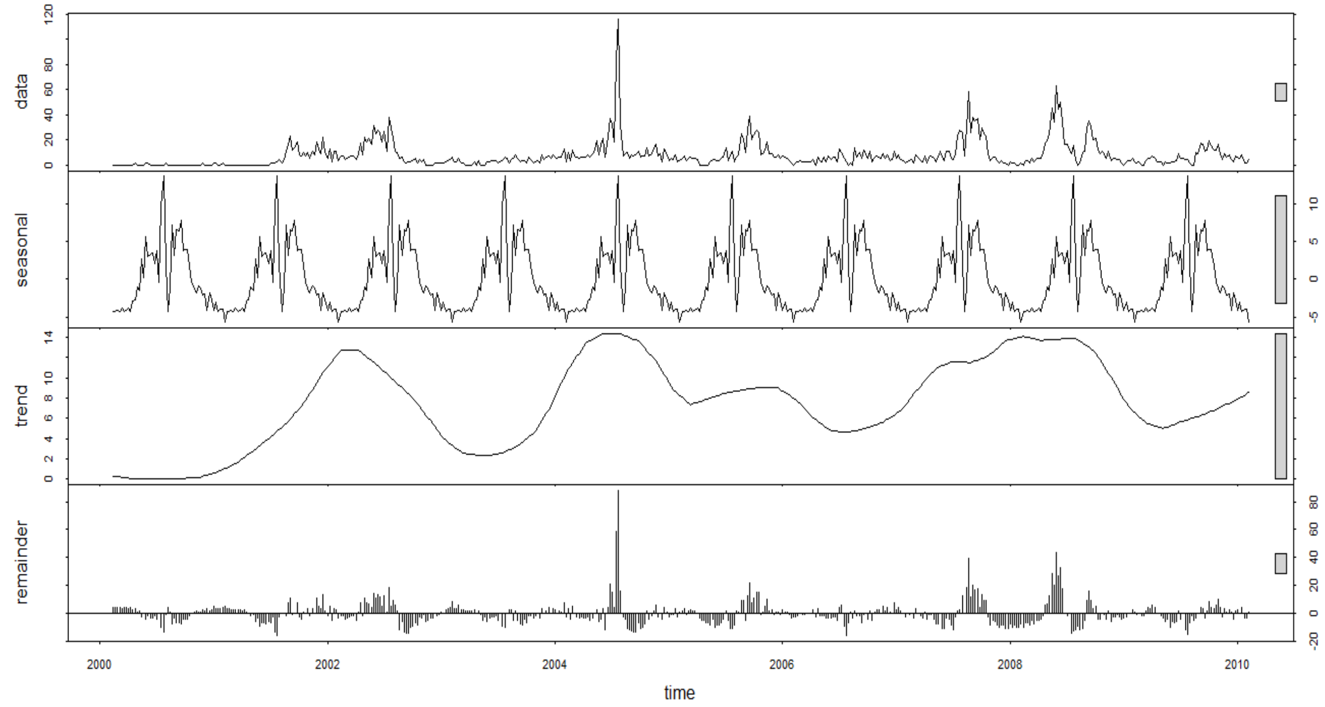
```
#formulating the time series:
sj.series=ts(sj.train$total_cases, frequency=52, start=c(1990,30,04))
iq.series=ts(iq.train$total_cases, frequency=52, start=c(2000,07,01))
```



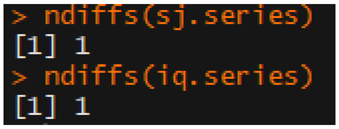
Act plot:



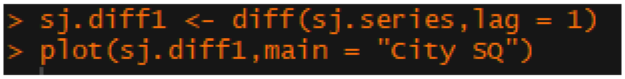
From the Acf plot we can see that our data is not stationary. To look at the seasonal and trend components we use Stl function to decompose our time series data.



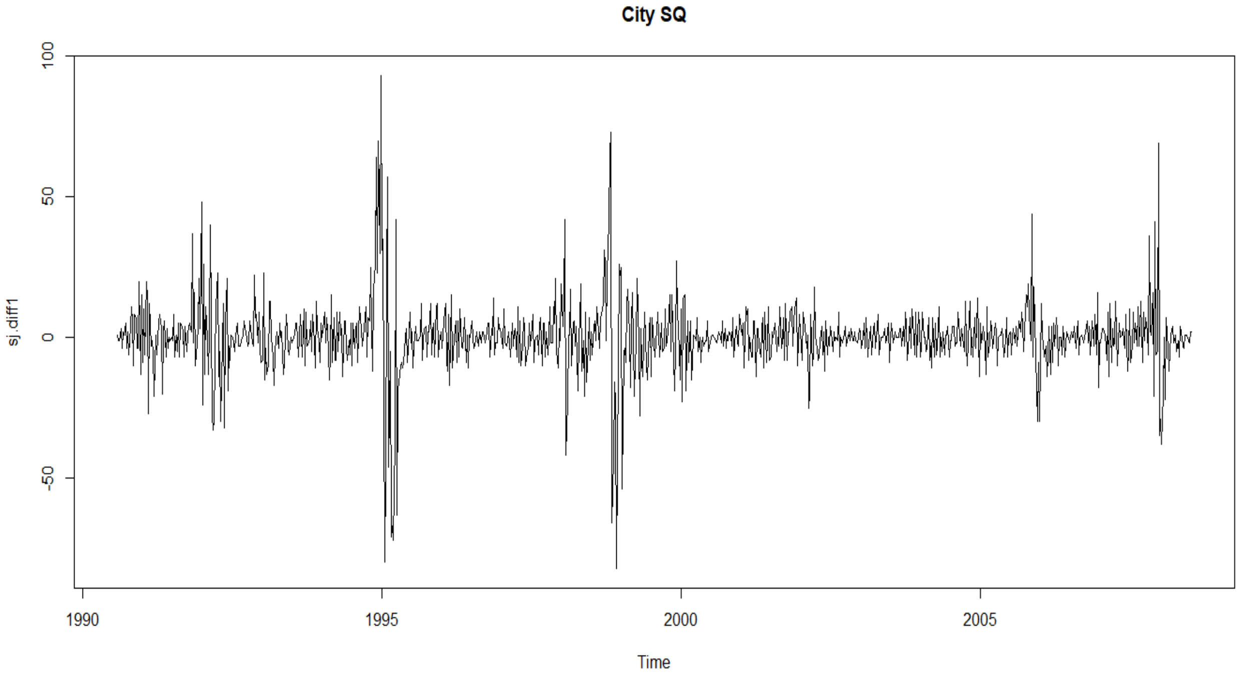
To apply the Arima model we must remove the trend and seasonal component. So, we will have to do differencing to our data. Ndiffs function gives us the number of times differencing should be done to make the data stationary.

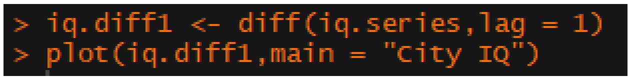


```
> ndiffs(sj.series)
[1] 1
> ndiffs(iq.series)
[1] 1
```

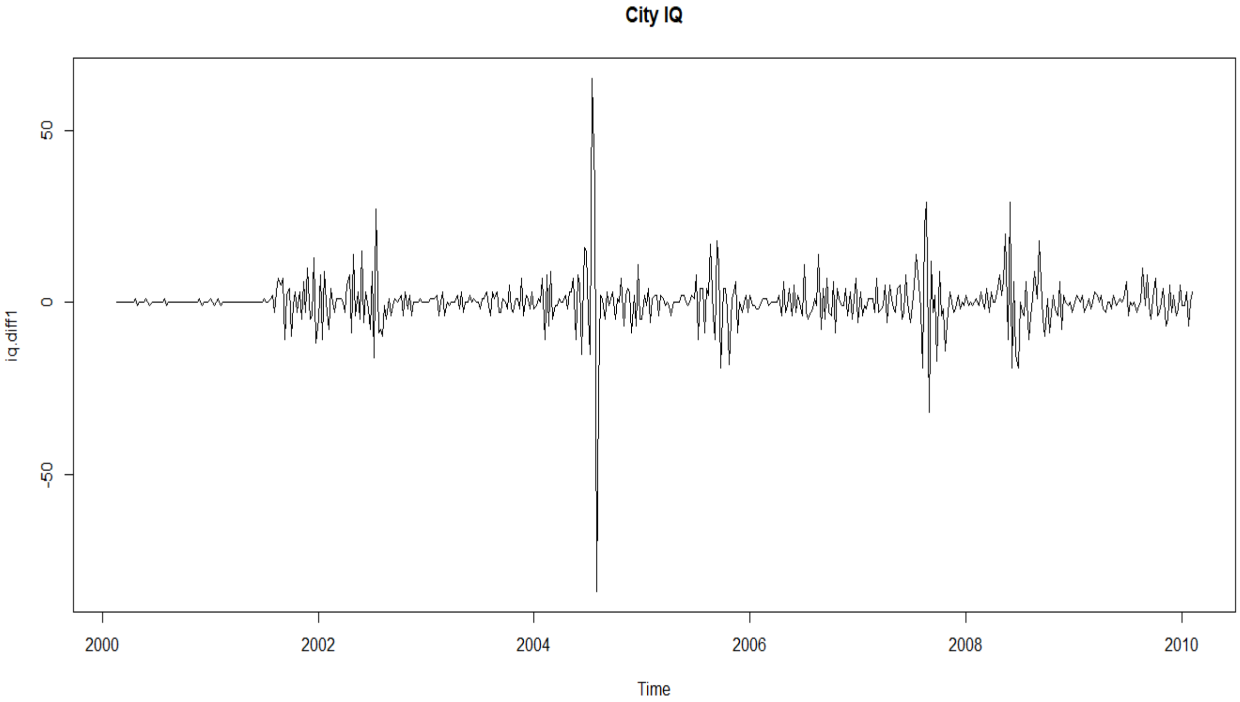


```
> sj.diff1 <- diff(sj.series,lag = 1)
> plot(sj.diff1,main = "City SQ")
```





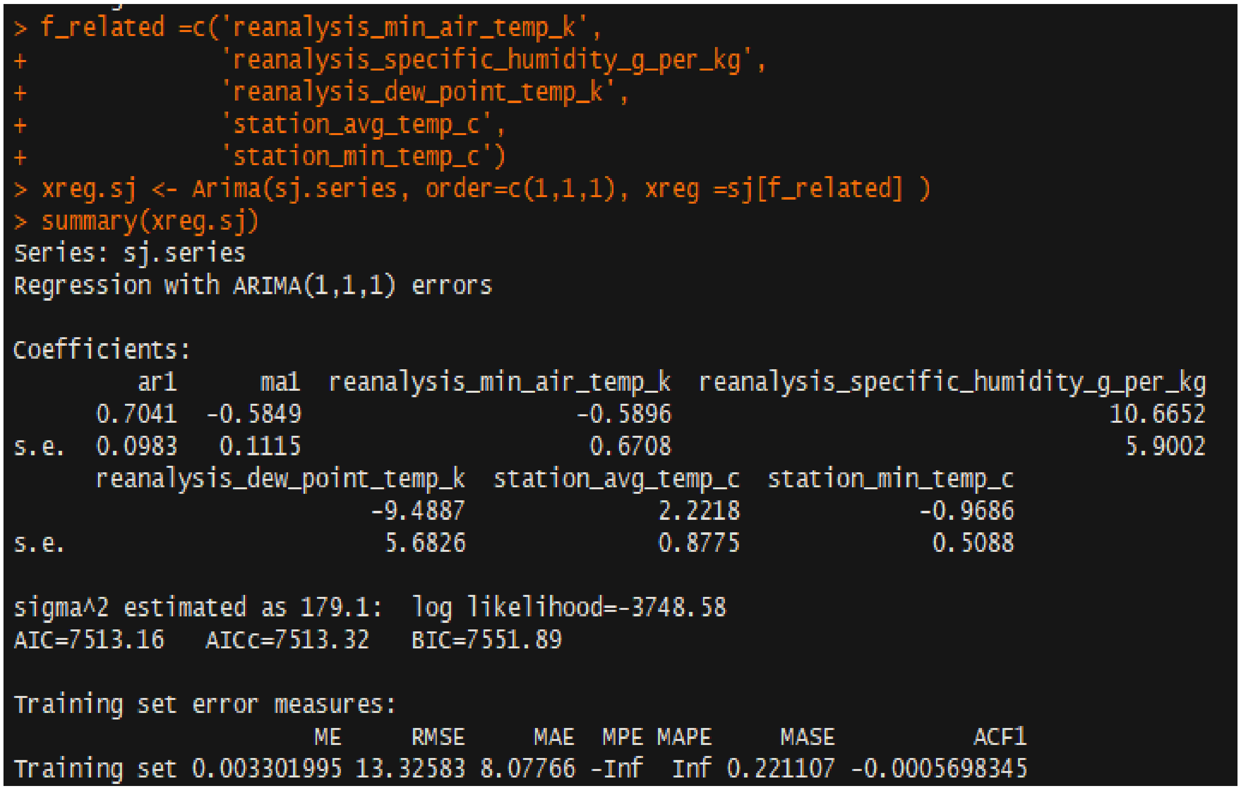
```
> iq.diff1 <- diff(iq.series,lag = 1)
> plot(iq.diff1,main = "City IQ")
```



Now we can see that the data is stationary, now our data is ready to apply the arima model.

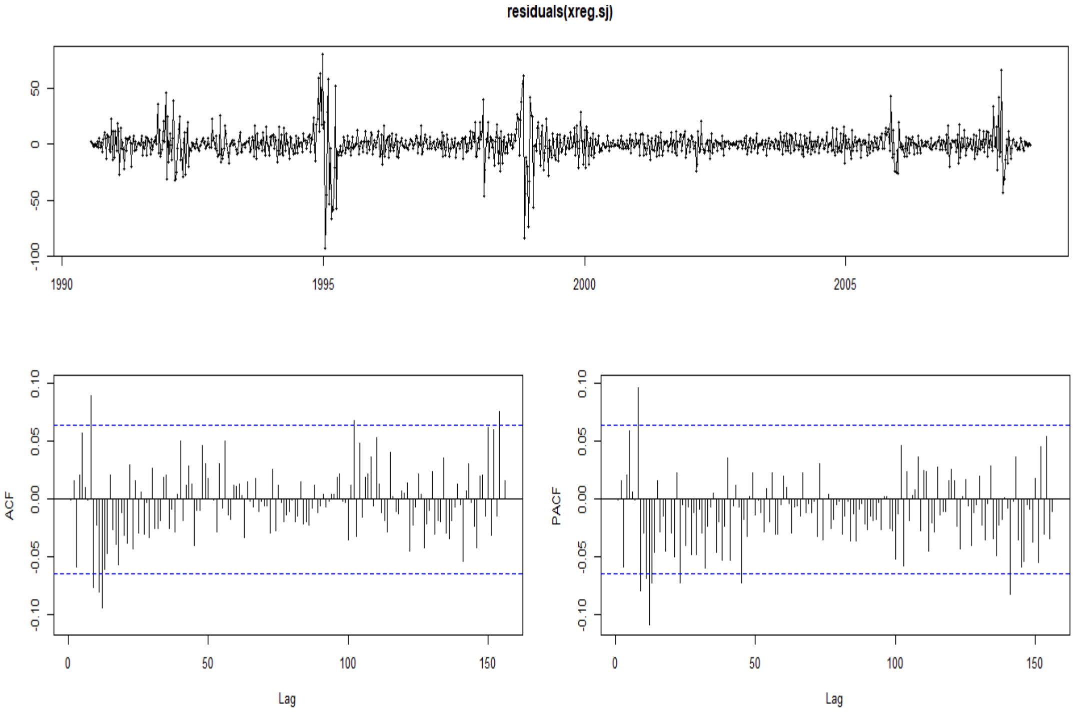
5.2.1 Xreg Arima

Xreg Arima for SJ City:

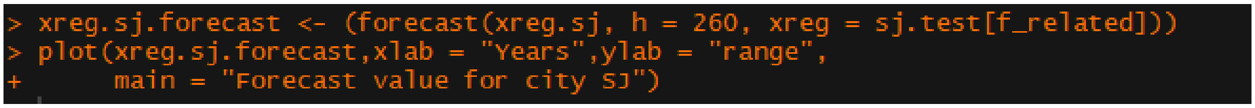


```
> f_related =c('reanalysis_min_air_temp_k',
+              'reanalysis_specific_humidity_g_per_kg',
+              'reanalysis_dew_point_temp_k',
+              'station_avg_temp_c',
+              'station_min_temp_c')
> xreg.sj <- Arima(sj.series, order=c(1,1,1), xreg =sj[f_related] )
> summary(xreg.sj)
Series: sj.series
Regression with ARIMA(1,1,1) errors

Coefficients:
         ar1      ma1  reanalysis_min_air_temp_k  reanalysis_specific_humidity_g_per_kg
      0.7041  -0.5849                    -0.5896                                10.6652
s.e.  0.0983   0.1115                     0.6708                                 5.9002
      reanalysis_dew_point_temp_k  station_avg_temp_c  station_min_temp_c
                          -9.4887              2.2218             -0.9686
s.e.                       5.6826              0.8775              0.5088

sigma^2 estimated as 179.1:  log likelihood=-3748.58
AIC=7513.16   AICc=7513.32   BIC=7551.89

Training set error measures:
                      ME     RMSE     MAE  MPE MAPE     MASE          ACF1
Training set 0.003301995 13.32583 8.07766 -Inf  Inf 0.221107 -0.0005698345
```

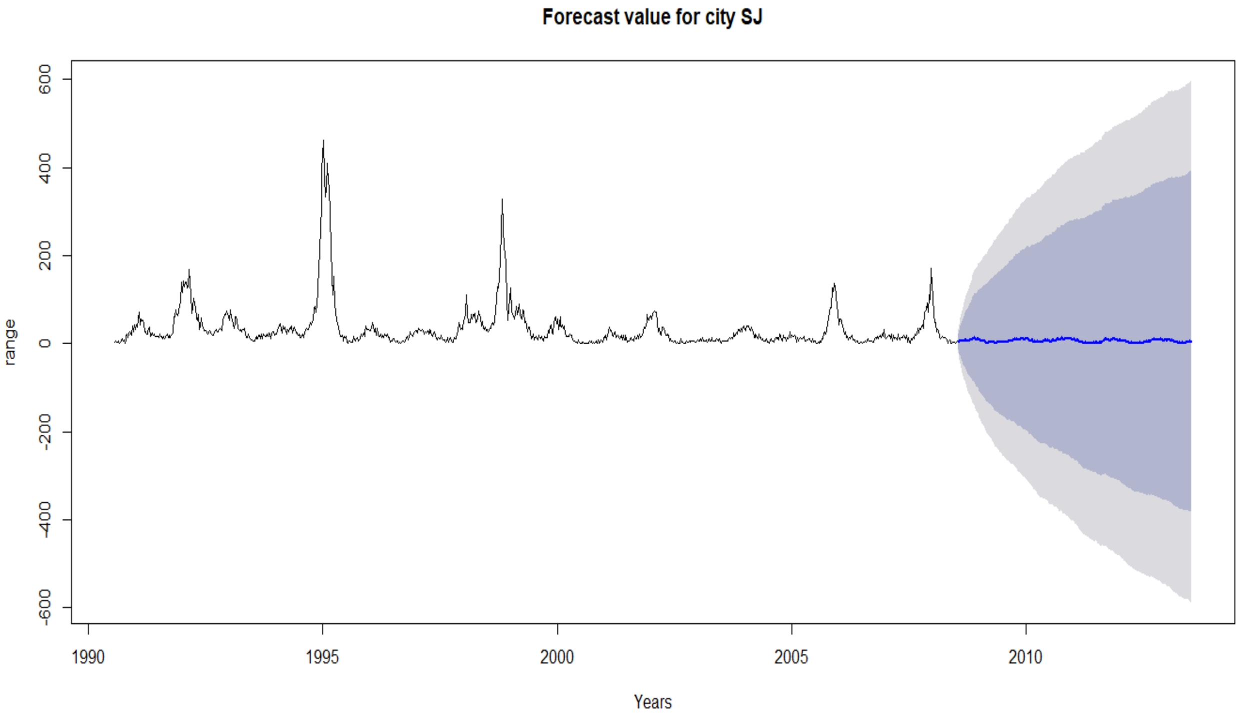
Residual Plot:



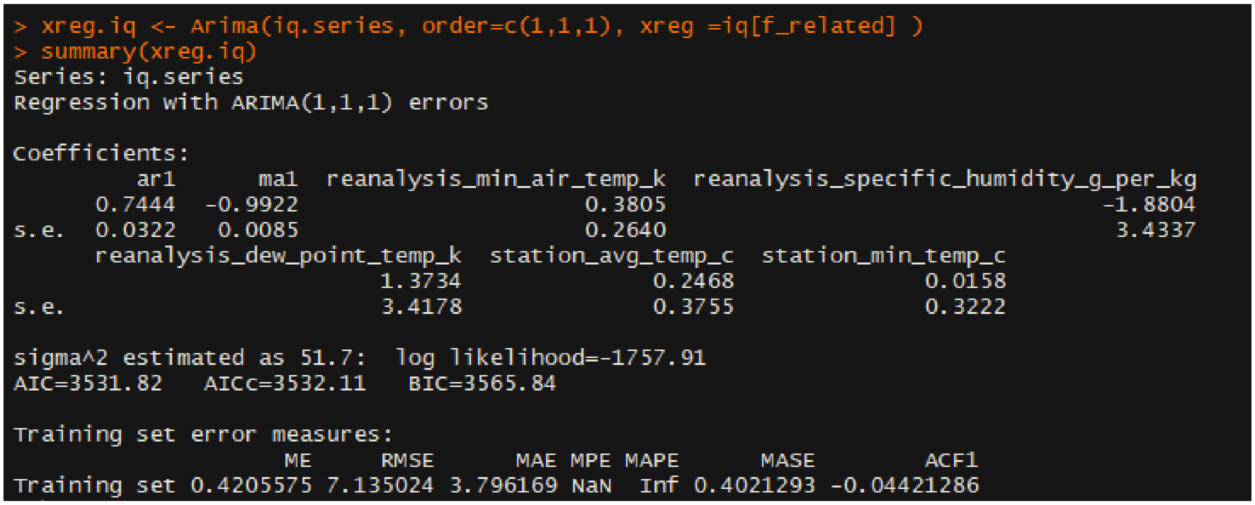
Forecasting the SJ test data:



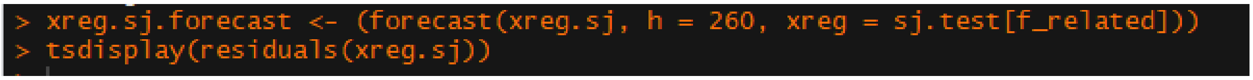
```
> xreg.sj.forecast <- (forecast(xreg.sj, h = 260, xreg = sj.test[f_related]))
> plot(xreg.sj.forecast,xlab = "Years",ylab = "range",
+      main = "Forecast value for city SJ")
```



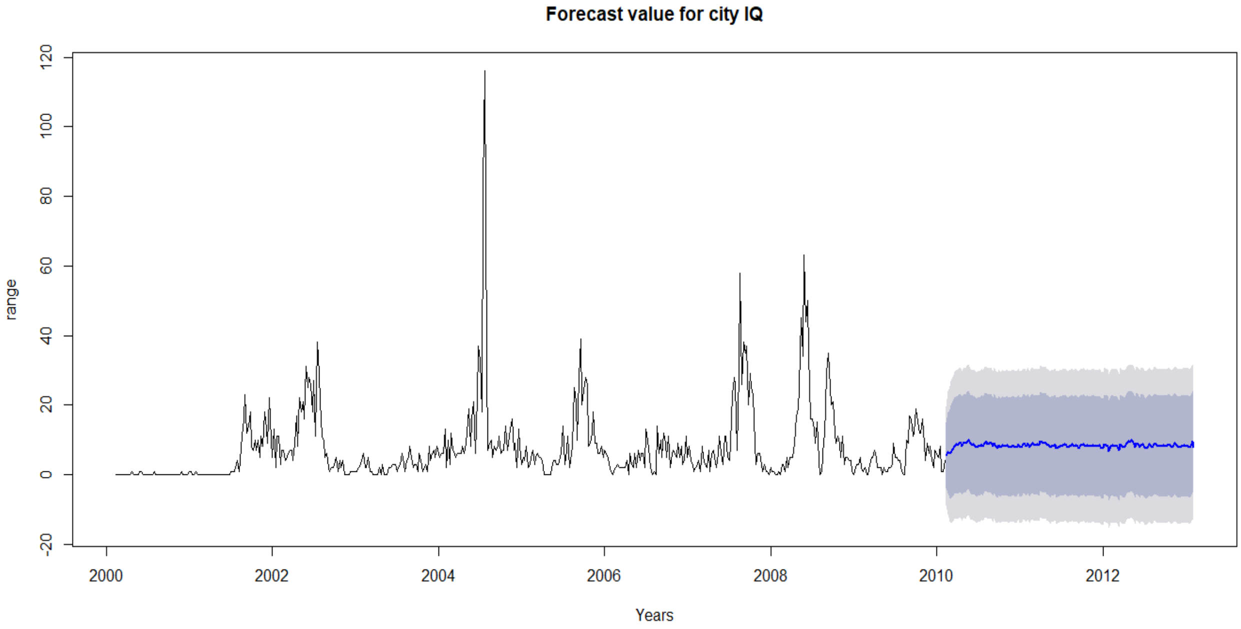
Xreg Arima for IQ City:



```
> xreg.iq <- Arima(iq.series, order=c(1,1,1), xreg =iq[f_related] )
> summary(xreg.iq)
Series: iq.series
Regression with ARIMA(1,1,1) errors

Coefficients:
         ar1      ma1  reanalysis_min_air_temp_k  reanalysis_specific_humidity_g_per_kg
      0.7444  -0.9922                     0.3805                                -1.8804
s.e.  0.0322   0.0085                     0.2640                                 3.4337
      reanalysis_dew_point_temp_k  station_avg_temp_c  station_min_temp_c
                           1.3734              0.2468              0.0158
s.e.                       3.4178              0.3755              0.3222

sigma^2 estimated as 51.7:  log likelihood=-1757.91
AIC=3531.82   AICc=3532.11   BIC=3565.84

Training set error measures:
                    ME     RMSE      MAE MPE MAPE      MASE        ACF1
Training set 0.4205575 7.135024 3.796169 NaN  Inf 0.4021293 -0.04421286
```

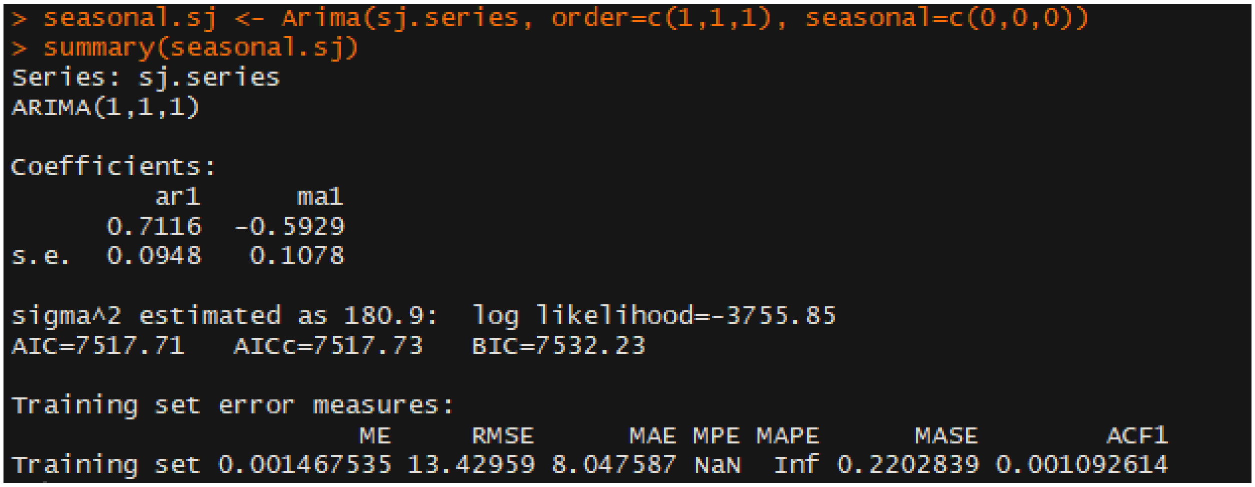


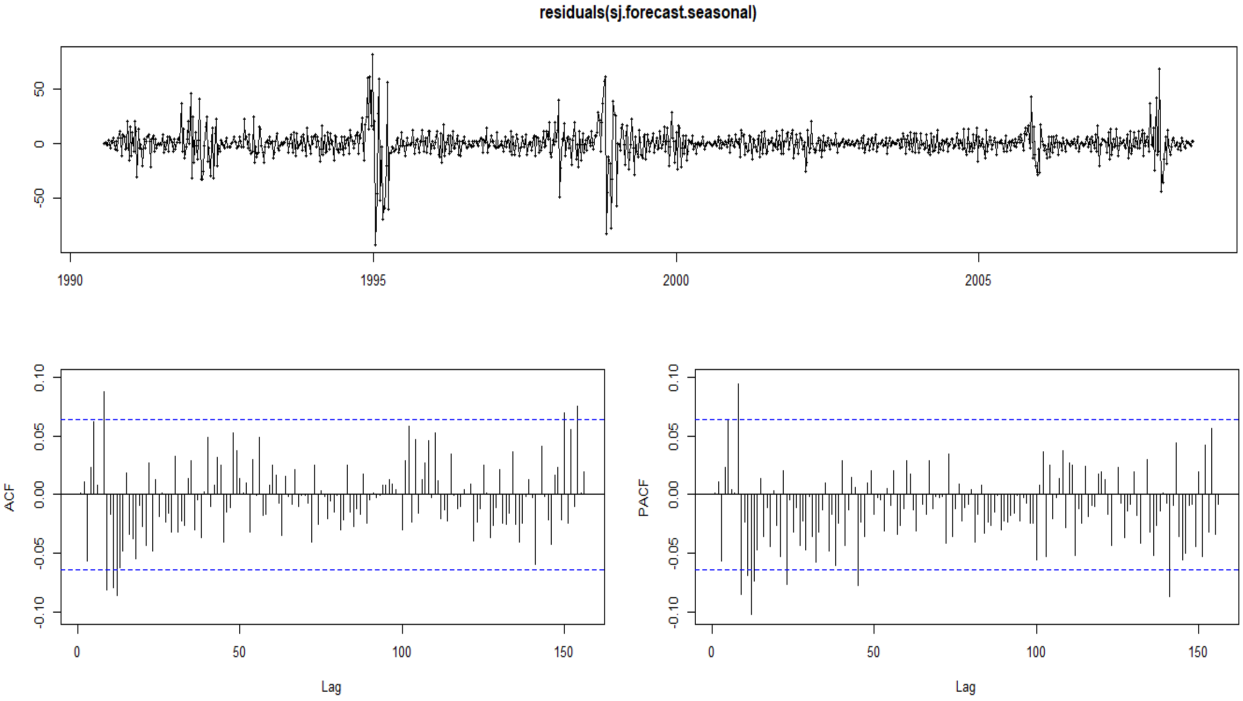
```
> xreg.sj.forecast <- (forecast(xreg.sj, h = 260, xreg = sj.test[f_related]))
> tsdisplay(residuals(xreg.sj))
```

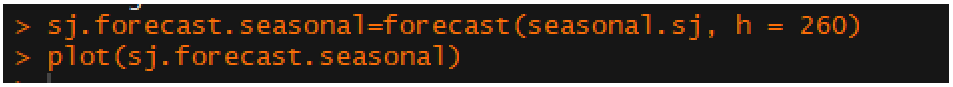


5.2.2 Seasonal Arima

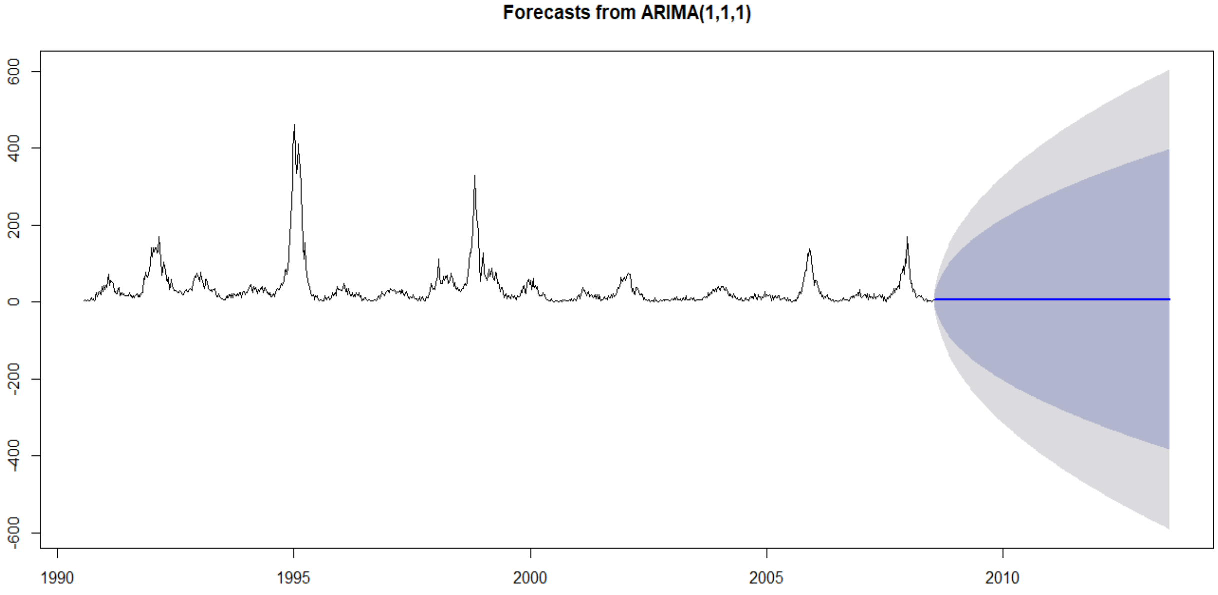
For sj City


```
> seasonal.sj <- Arima(sj.series, order=c(1,1,1), seasonal=c(0,0,0))
> summary(seasonal.sj)
Series: sj.series
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.7116  -0.5929
s.e.  0.0948   0.1078

sigma^2 estimated as 180.9:  log likelihood=-3755.85
AIC=7517.71   AICc=7517.73   BIC=7532.23

Training set error measures:
                      ME     RMSE      MAE MPE MAPE      MASE         ACF1
Training set 0.001467535 13.42959 8.047587 NaN  Inf 0.2202839 0.001092614
```

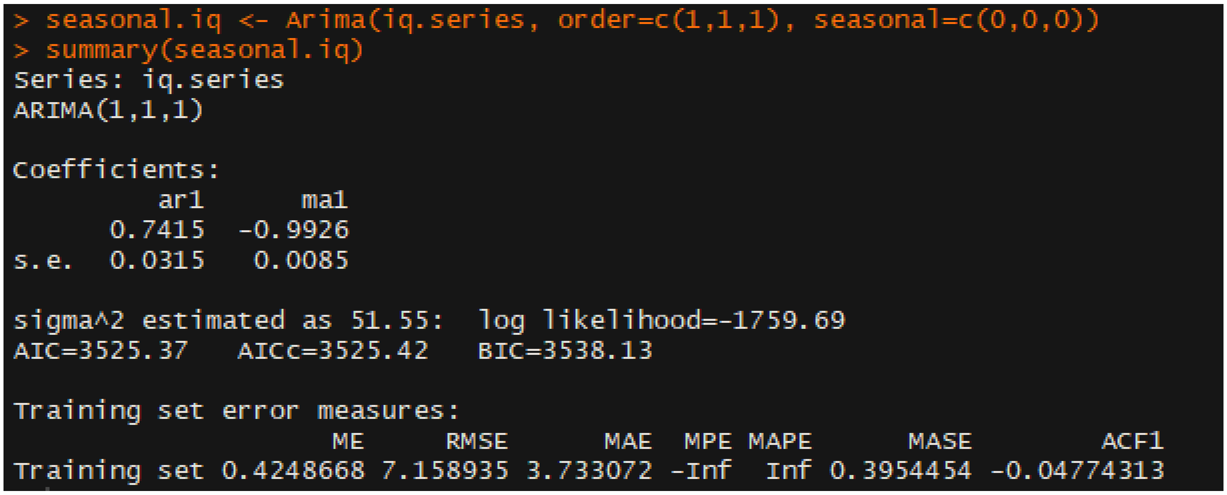




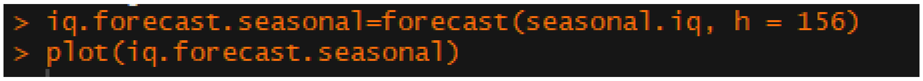
```
> sj.forecast.seasonal=forecast(seasonal.sj, h = 260)
> plot(sj.forecast.seasonal)
```



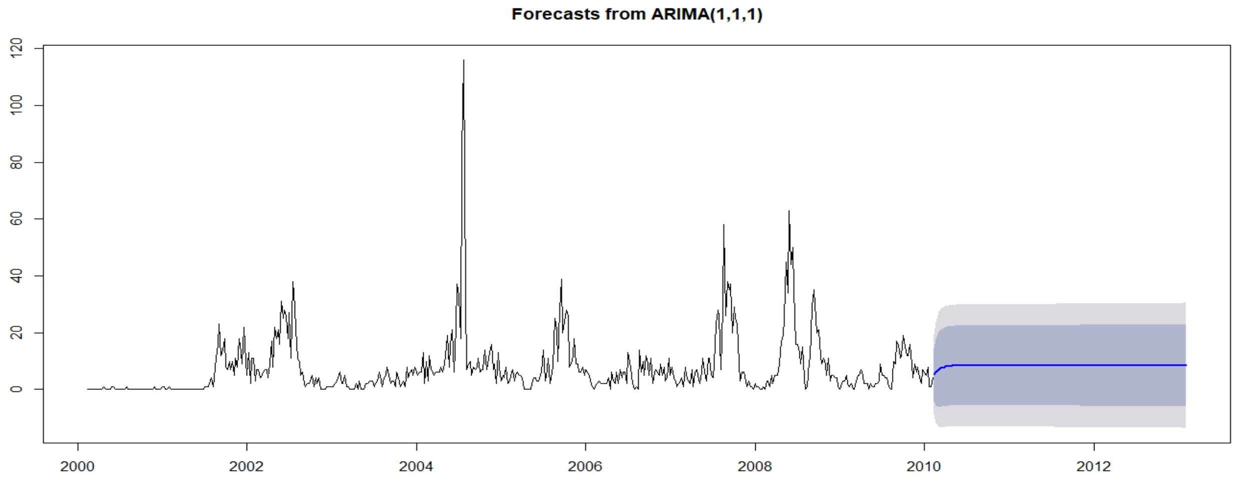
For IQ city:



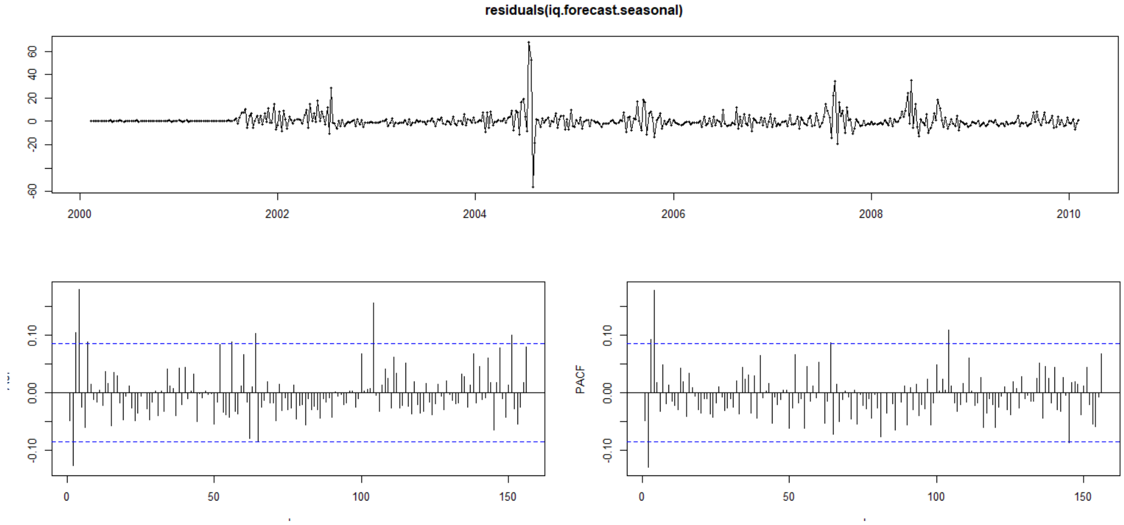
```
> seasonal.iq <- Arima(iq.series, order=c(1,1,1), seasonal=c(0,0,0))
> summary(seasonal.iq)
Series: iq.series
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.7415  -0.9926
s.e.  0.0315   0.0085

sigma^2 estimated as 51.55:  log likelihood=-1759.69
AIC=3525.37   AICc=3525.42   BIC=3538.13

Training set error measures:
                    ME     RMSE      MAE  MPE MAPE      MASE        ACF1
Training set 0.4248668 7.158935 3.733072 -Inf  Inf 0.3954454 -0.04774313
```



```
> iq.forecast.seasonal=forecast(seasonal.iq, h = 156)
> plot(iq.forecast.seasonal)
```



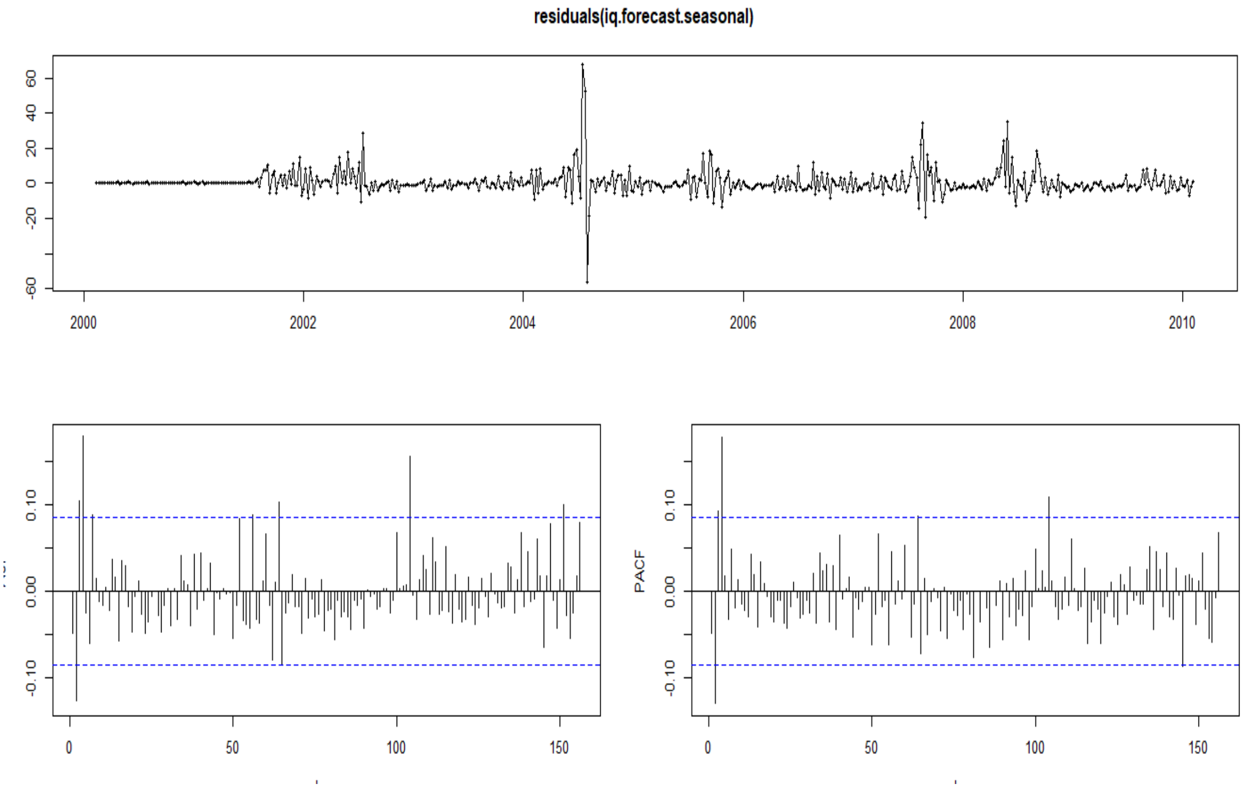
Residue plot:

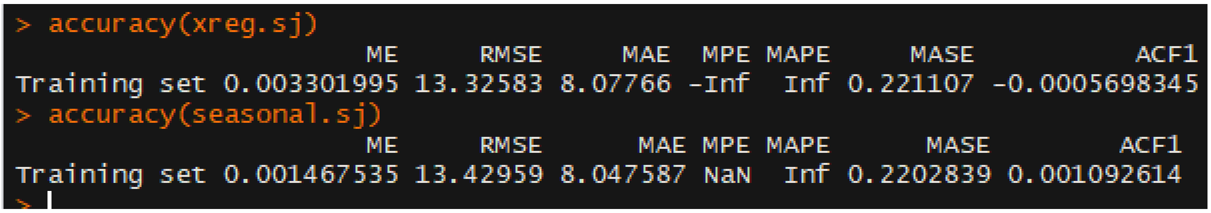


5.2.3 Assessment of the model

Accuracy test

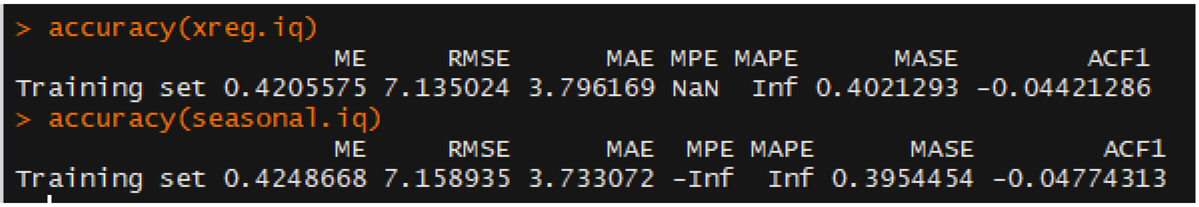
For City SJ:





```
> accuracy(xreg.sj)
                       ME     RMSE     MAE  MPE MAPE     MASE          ACF1
Training set 0.003301995 13.32583 8.07766 -Inf  Inf 0.221107 -0.0005698345
> accuracy(seasonal.sj)
                       ME     RMSE      MAE MPE MAPE      MASE        ACF1
Training set 0.001467535 13.42959 8.047587 NaN  Inf 0.2202839 0.001092614
>
```

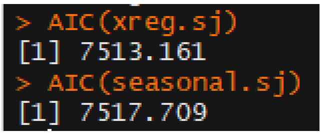
For City IQ:



```
> accuracy(xreg.iq)
                     ME     RMSE      MAE MPE MAPE      MASE        ACF1
Training set 0.4205575 7.135024 3.796169 NaN  Inf 0.4021293 -0.04421286
> accuracy(seasonal.iq)
                     ME     RMSE      MAE  MPE MAPE      MASE        ACF1
Training set 0.4248668 7.158935 3.733072 -Inf  Inf 0.3954454 -0.04774313
```

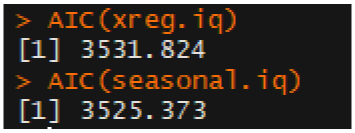
AIC (Akaike information criterion):

For City SJ:



```
> AIC(xreg.sj)
[1] 7513.161
> AIC(seasonal.sj)
[1] 7517.709
```

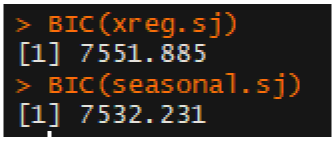
For City IQ:



```
> AIC(xreg.iq)
[1] 3531.824
> AIC(seasonal.iq)
[1] 3525.373
```

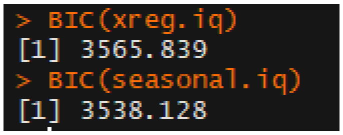
BIC (Bayesian information criterion)

For City SJ:



```
> BIC(xreg.sj)
[1] 7551.885
> BIC(seasonal.sj)
[1] 7532.231
```

For City IQ:



```
> BIC(xreg.iq)
[1] 3565.839
> BIC(seasonal.iq)
[1] 3538.128
```

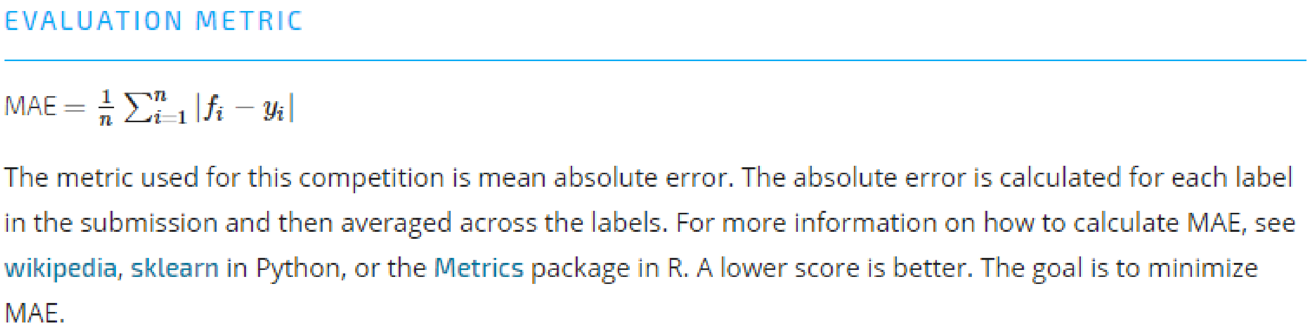
From the accuracy, AIC and BIC tests we can see that the seasonal arima has better results when compared to the xreg arima model.

6 Discussion and Recommendation

We have applied the above discussed predictive techniques to the data and then tested the models. Later, these models were used for test data and the obtained output in each of the cases is stored in a .csv file. This output file is like the submission format mentioned in the project competition website.

Evaluation Metric:

The Evaluation metric for this competition is MAE (Mean Absolute Error) explained in detail below:

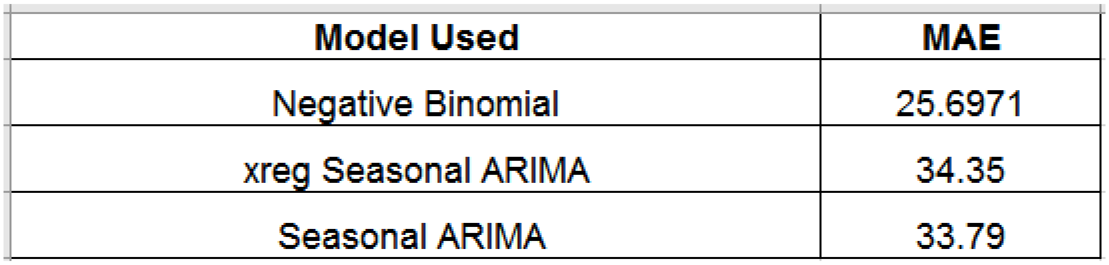


EVALUATION METRIC

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

The metric used for this competition is mean absolute error. The absolute error is calculated for each label in the submission and then averaged across the labels. For more information on how to calculate MAE, see wikipedia, sklearn in Python, or the Metrics package in R. A lower score is better. The goal is to minimize MAE.

Out of the models used by us for the prediction, we see that the **Negative Binomial model** we used has low MAE and thus performed better when the output is uploaded to the competition website.



Model Used	MAE
Negative Binomial	25.6971
xreg Seasonal ARIMA	34.35
Seasonal ARIMA	33.79

Thus, we recommend using the Negative Binomial model for the prediction of total_cases of dengue per week. in case of this project.

Discussion and interpretation of the results:

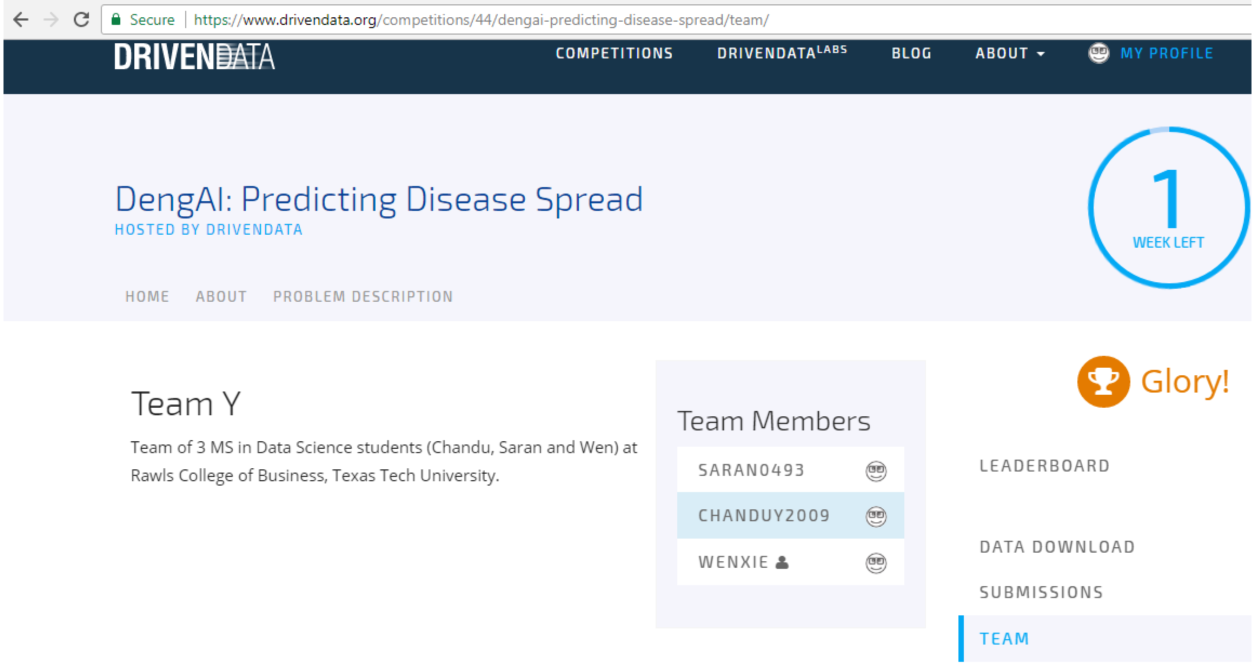
Because dengue fever is a tropical disease, we would expect it to be more popular in place with high temperature, high precipitation and thus high humidity. The disease is transmitted by mosquitoes, whose peak season is summer, so we will expect summer to have more dengue cases than the rest of the year.

Iquitos is a Peruvian port city and surrounded by green spaces and water sources, it is an ideal environment for mosquitoes. San juan, in the other hand, is a city on an island (more isolated) and has a much lower population density than Iquitos. All these information's suggest that we will need 2 separate models, one for each city, which we have done.

The correlation strengths differ for each city, but it looks like reanalysis_specific_humidity_g_per_kg and reanalysis_dew_point_temp_k are the most strongly correlated with total_cases. This makes sense: We know mosquitos thrive wet climates, the wetter the better! As we all know, "cold and humid" is not a thing. So, it's not surprising that as minimum temperatures, maximum temperatures, and average temperatures rise, the total_cases of dengue fever tend to rise as well. Interestingly, the precipitation measurements bear little to no correlation to total_cases, despite strong correlations to the humidity measurements, as evident by the heatmaps above.

7 Submission to Competition Website

As suggested, we have registered as **TeamY** in this project competition on drivendata.org:



DengAI: Predicting Disease Spread

HOSTED BY DRIVENDATA

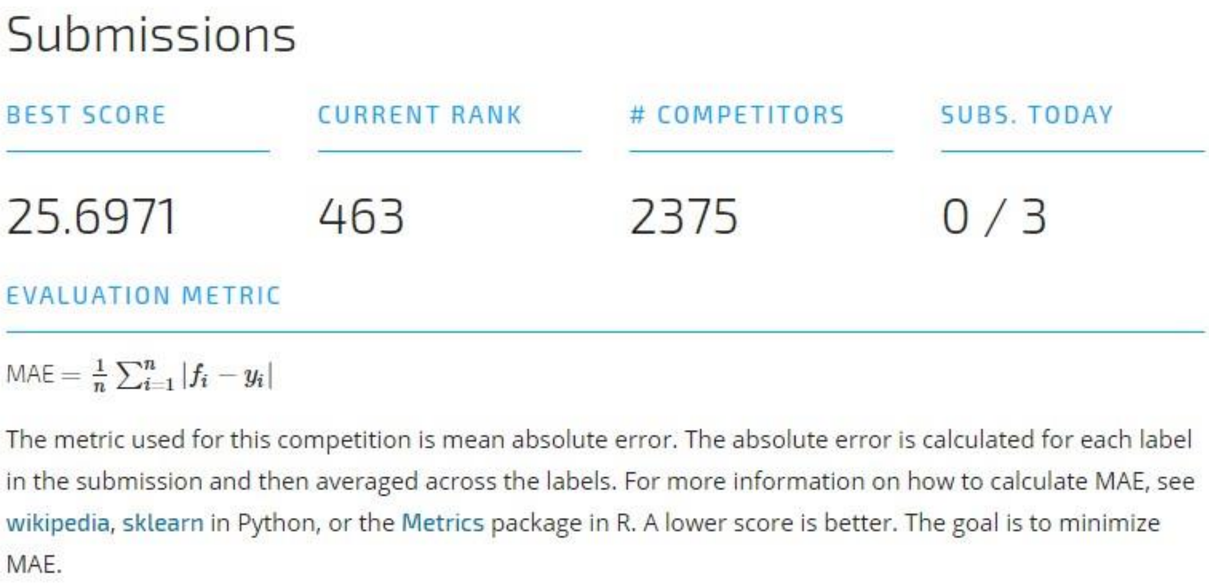
Team Y

Team of 3 MS in Data Science students (Chandu, Saran and Wen) at Rawls College of Business, Texas Tech University.

Team Members

- SARAN0493
- CHANDUY2009
- WENXIE

And, we submitted the output file to the competition website and below is the screenshot of the submission.



Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
25.6971	463	2375	0 / 3

EVALUATION METRIC

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

The metric used for this competition is mean absolute error. The absolute error is calculated for each label in the submission and then averaged across the labels. For more information on how to calculate MAE, see wikipedia, sklearn in Python, or the Metrics package in R. A lower score is better. The goal is to minimize MAE.