Noé Flores
Predict Dengue Fever

Introduction:

When we stop to consider the creatures that pose the biggest threat to our health, few would immediately recognize the mosquito to be the most dangerous. Mosquitoes are in fact one of, if not the most dangerous creatures to humanity because of their ability to spread and transmit deadly disease is unrivaled. The CDC estimates that mosquitoes kill over one million people worldwide per year just through the spread of malaria, which is just one of the many diseases they are capable of spreading to the human populace.

Dengue fever is another mosquito-borne disease that occurs in tropical and sub-tropical parts of the world. In severe cases, dengue fever can cause severe bleeding, low blood pressure, and like malaria, death. Dengue fever has spread across the globe. It was once prevalent in South Asia and the Pacific Islands but has recently found a home in Latin America, which is now prone to nearly five hundred million cases per year. Two locations in Latin America that present favorable climate conditions for mosquitoes to thrive and spread Dengue fever are San Juan, Puerto Rico, a United States territory, and Iquitos, Peru.

The problem presented for us is to build a predictive model 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—to help us forecast the number of dengue fever cases reported each week in San Juan, Puerto Rico and Iquitos, Peru. We will perform an exploratory analysis of our data set to gather information regarding the variables we have available to incorporate into this research and follow up with any data preparation needed to ensure that we are working with optimal data. Finally, we will build and compare several models to determine which has the most significant predictive value, interpretability, and practicality with respect to our goal of identifying the number of Dengue cases reported each week.

Data Analysis and Preparation:

Our data set contained two files. Our first file included 1456 observations of 24 different variables serving as possible predictors of Dengue cases. The second file in our analysis contains the labels of the observations, such as the cases of Dengue fever per week and per city. The first thing we will do is merge these files together producing a training set of 1456 observations and 25 different variables. The table provides a preview of some of the variables we will use during the analysis.

Figure 1: Variables and Descriptions.

Dengue Fever Data	**Contents**
Total Observations	1456
Variable Preview	**25 total Variables**
City	
Year	
Precipitation_amt_mm	
Reanalysis_air_temp_k	
Reanalysis_avg_temp_k	
Total_cases	

It is common to find missing or erroneous values in large data sets. Inaccurate or outlying data can be dealt with in a number of different ways or one we can simply allow those data points into the model. Missing values must be imputed before moving forward with the process. All of the predictor variables had missing values, which ranged anywhere from 10 to 194. To address these missing values, we merely imputed the variable with variable's mean(µ) value. We can now look at some measures of central tendency and verify that any missing values have been imputed correctly.

Figu re 2: Measures of central Tendency

Variable	N	Min	Max	Mean	Median	St.Dev
ndvi_ne	1,456	-0.41	0.51	0.14	0.14	0.13
ndvi_nw	1,456	-0.46	0.45	0.13	0.13	0.12
ndvi_se	1,456	-0.02	0.54	0.20	0.20	0.07
ndvi_sw	1,456	-0.06	0.55	0.20	0.19	0.08
precipitation_amt_mm	1,456	0.00	390.60	45.76	38.71	43.52
reanalysis_air_temp_k	1,456	294.64	302.20	298.70	297.70	1.36
reanalysis_avg_temp_k	1,456	294.89	302.93	299.22	299.30	1.26
reanalysis_dew_point_temp_k	1,456	289.64	298.45	295.25	295.60	1.52
reanalysis_max_air_temp_k	1,456	297.80	314.00	303.43	302.50	3.22
reanalysis_min_air_temp_k	1,456	286.90	299.90	295.72	296.20	2.56
reanalysis_precip_amt_kg_per_m2	1,456	0.00	570.50	40.15	27.37	43.29
reanalysis_relative_humidity_percent	1,456	57.79	98.61	82.16	80.37	7.13
reanalysis_sat_precip_amt_mm	1,456	0.00	390.60	45.76	38.71	43.52
reanalysis_specific_humidity_g_per_kg	1,456	11.72	20.46	16.48	17.07	1.54
reanalysis_tdtr_k	1,456	1.36	16.03	4.90	2.86	3.53
station_avg_temp_c	1,456	21.40	30.80	27.19	27.39	1.27
station_diur_temp_rng_c	1,456	4.53	15.80	8.06	7.39	2.10
station_max_temp_c	1,456	26.70	42.20	32.45	32.80	1.95
station_min_temp_c	1,456	14.70	25.60	22.10	22.20	1.57
station_precip_mm	1,456	0.00	543.30	39.33	24.45	47.10
total_cases	1,456	0.00	461.00	24.68	12.00	43.60

The dataset contains two distinct locals where we want to predict the spread of Dengue. We decided it was best to split the data into separate files. In the table below we can see a breakdown of the total cases of Dengue in each of those two cities. It's clear there is a large spread in the number of cases.

Figure 3: Total Cases of Dengue fever:

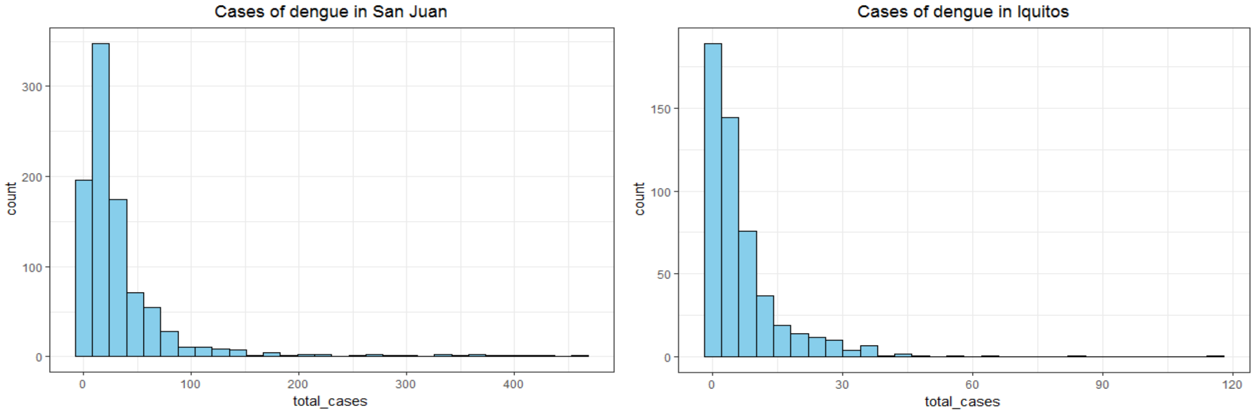
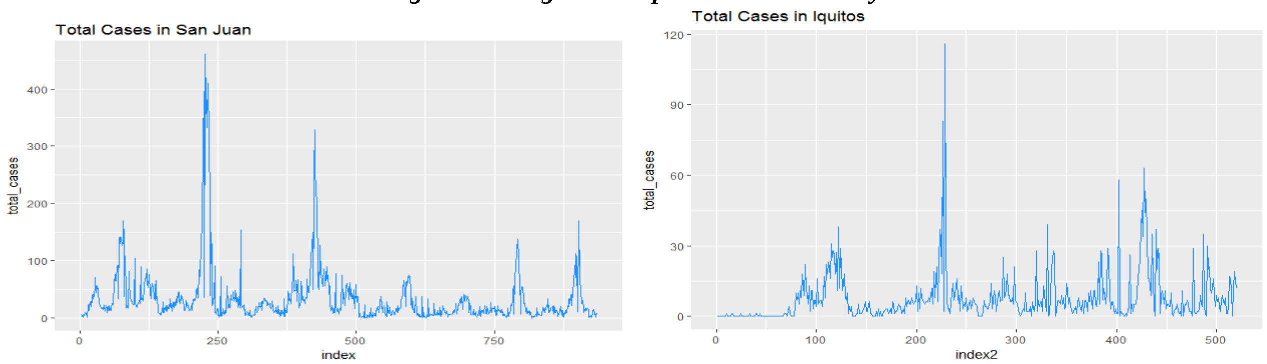
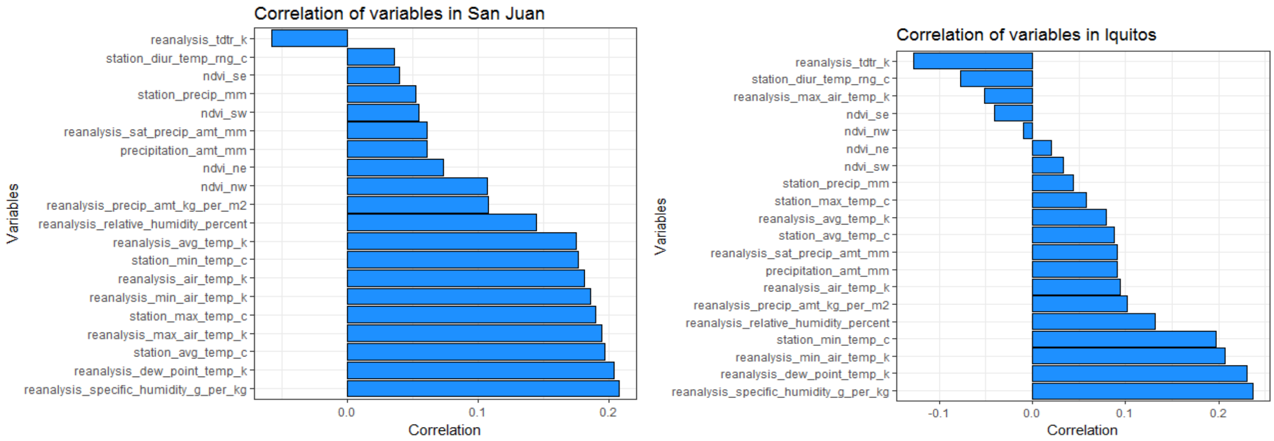


Figure 4: Dengue cases per week in each city.

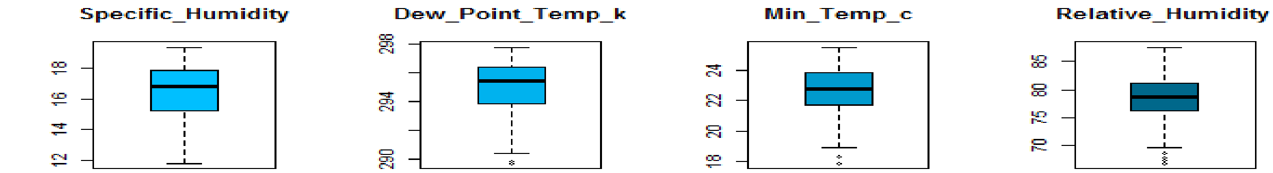


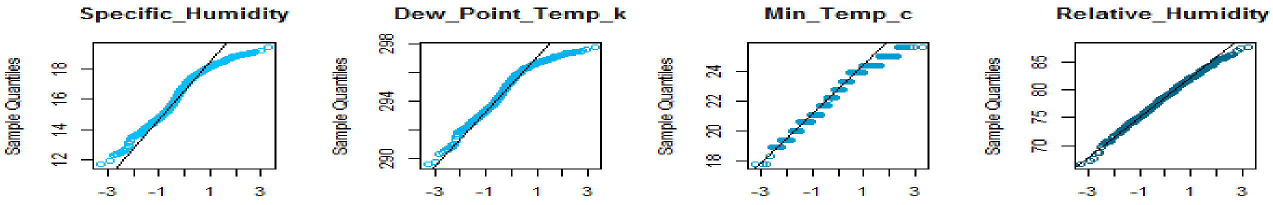
The time series plots above give us a different view of the Dengue fever outbreaks throughout the course of the data period. The time periods on the bottom of the plot represent weeks. We can see a bit more clearly the instances whre a substantial outbreak of Dengue occurred and other periods of relatively moderate and even zero outbreaks.

Figure 5: Box-plots of correlations per city;



The correlations above present an interesting picture. In both cities, it would appear that reanalysis_specific_humidity_g_per_kg and reanalysis_dew_point_temp_k have the strongest correlation to total cases. There is also a strong correlation factor associated with the temperature in both localities, although admittedly the variables aren't uniform in each case justifying separation.





Model Research and Literature:
In this particular analysis, we are dealing with a situation where we have to predict the number of times an event will happen. The literature listed below contains information relative to our task of predicting the number of future diagnosed Dengue fever cases per week. Each journal piece contains examples of the models we elected to use in developing predictive models for count data and therefore we are confident that we will find a decent fitting model. Since we wanted to be as thorough as possible, we decided to implement five different models. After researching the literature, we will proceed with a Linear regression model, Negative Binomial regression model, Neural Net model, Arima model, and a Random Forest regression model.

1. Davis, R. A., & Wu, R. (2009). *A negative binomial model for time series of counts.*
2. Bergh, F. v. d., Holloway, J. P., Pienaar, M., Koen, R., Elphinstone, C. D., & Woodborne, S. (2008). *A comparison of various modeling approaches applied to cholera case data.*
3. Wang, J., & Deng, Z. (2016). *Modeling and prediction of oyster norovirus outbreaks along gulf of mexico coast. Environmental Health Perspectives.*
4. Artola, C., Pinto, F., & de, P. G. (2015). *Can internet searches forecast tourism inflows? International Journal of.*
5. Sharma, V. C., Frankenfield, D., Gupta, A., & Singh, R. K. (2015). *Ensemble approach for zoonotic disease forecasting using machine learning techniques. International Journal of Business Analytics and Intelligence.*

Modeling Building:
Our work will center on building four distinct models. Two model will be simple linear regression models, and the other two will be logistic regression models. These models will also differ in their composition and variable selection.

Model 1 linear Regression
model1 <- lm(formula = total_cases ~ mtrain(all variables) Stepwise selection

Model 1 is a simple linear regression model incorporating the original variables utilizing stepwise selection

Model 2Negative Binomial Regression
model2 <-glm.nb(formula = total_cases ~ mtrain(all variables) Stepwise selection

Model 2 is a Negative Binomial regression model incorporating the original variables utilizing stepwise selection

Model 3 Neural Net Regression
model3 <-nnet(formula = total_cases ~ mtrain(all variables)

Model 3 Neural Net regression model incorporating all the original variables at the time of analysis.

Model 4 Arima Regression Model
model4 <- fit <- auto.arima(sjts) + fit_iq <- auto.arima(iqts) Auto arima model for each individual locality

Model 4 is an auto.arima model forecasting dengue out breaks individually in each city

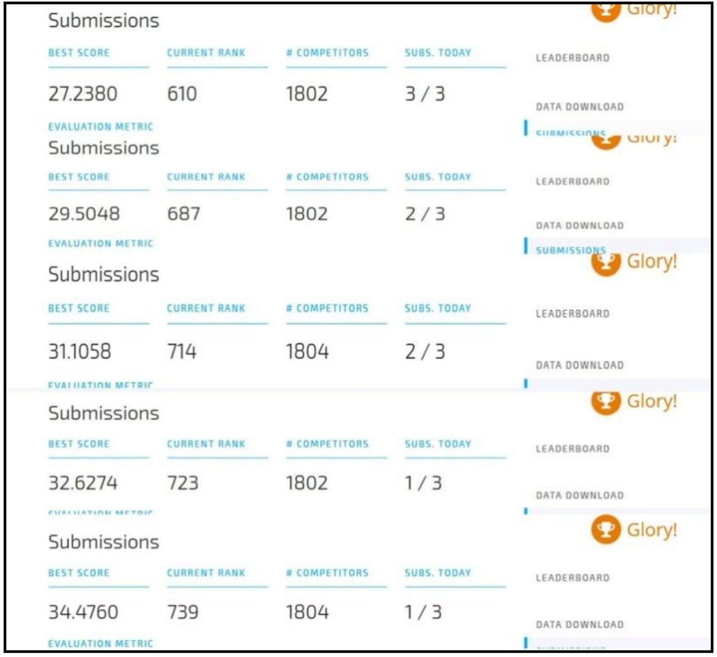
Model 5 Random Forest Regression Model
Model5 <- sj_rf <- randomForest(total_cases ~ sj_train(select variable) + iq_rf <- randomForest(total_cases ~ iq_train(select variable) +

Model 5 is a random forest utilizing the strongest variables from the stepwise selection process in model 2

Modeling Selection:

Since we would have needed to split the training data set in order to get uniform fit metrics for our models, we decided to rely on the drivendata.org metric which is the mean absolute error (MAE). The figure below has a display of all five models ranked from best to worst.

Figure 6: Driven Data Results



Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
27.2380	610	1802	3 / 3

EVALUATION METRIC

Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
29.5048	687	1802	2 / 3

EVALUATION METRIC

Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
31.1058	714	1804	2 / 3

EVALUATION METRIC

Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
32.6274	723	1802	1 / 3

EVALUATION METRIC

Submissions

BEST SCORE	CURRENT RANK	# COMPETITORS	SUBS. TODAY
34.4760	739	1804	1 / 3

EVALUATION METRIC

Model	MAE
Random Forest - Model 5	27.2380
Negative Binomial - Model 2	29.5048
Neural Net -Model 3	31.1058
Linear Regression - Model 1	32.6274
Auto.Arima - Model 4	34.4760

The champion model sitting at the top of the list in Figure.6 was the Random Forest regression analysis, with a score of 27.2380. Given the strong scores we saw from the negative binomial regression model and the linear regression model, we were a bit surprised that the new modeling techniques performed so well.

Limitations:

The main drawback for me with this analysis was the data itself. I found it a bit difficult to work through which model would work best, should I split the data or run the model as a whole, does splitting the data work best in some cases and not others, etc. The data was also very robust and the variable identification was difficult to process by merely reading the variable name. It would appear to that a better description of the variables would be useful.

Future work and learning:

In a future analysis of this nature where we have a lot of variables to work through, I would look to expand on the more sophisticated modeling techniques, such as the Neural Networks and Random Forest models. These models performed surprisingly well with count data. With that said, I found it surprising that the tried and true linear regression and negative binomial regression models performed so well. As with every analysis, and given the ease at which models can be developed through statistical analysis packages, I think it's still true that we should build many models and have a set criteria for model selection. This should entail taking the time to separate the training data set in order to produce uniform fit statistics that are easily quantified. Finally, this wasn't my best work in terms of the score, so the learning is persistent and never-ending.

Appendix: R code

```
###413 Final dengAi

# Loading the data
train <- read.csv(file.path("C:/Users/Noe/Downloads/deng_features_train.csv"),sep=",")
test <- read.csv(file.path("C:/Users/Noe/Downloads/deng_features_test.csv"),sep=",")
labels <- read.csv(file.path("C:/Users/Noe/Downloads/deng_labels_train.csv"),sep=",")
submission_format <-
read.csv(file.path("C:/Users/Noe/Downloads/submission_format.csv"),sep=",")

###Start###
require(ggplot2)
require(forecast)
require(moments)
require(tidyverse)
require(magrittr)
require(randomForest)
require(zoo)
require(mlr)
require(nnet)
require(psych)
require(readr)
require(dplyr)
require(corrplot)
require(outliers)
require(rpart)
require(MASS)
require(Rmisc)
require(stargazer)
require(dplyr)
require(RColorBrewer)

###check my data train
str(train)
head(train)
tail(train)

###check my data train
str(test)
head(test)
tail(test)

###check my data train
str(labels)
head(labels)
tail(labels)
```

```
###Merge training and labels data
mtrain <- merge(train,labels,by=c("city","year","weekofyear"))

###check my mtrain
str(mtrain)
head(mtrain)
tail(mtrain)
summary(mtrain)

###missing values.
mv <- as.data.frame(apply(mtrain,2, function(x) any(is.na(x))))
colnames(mv) <- 'is_there_NA'
mv

colSums(is.na(mtrain))

###Impute with mean
for(i in 1:ncol(mtrain)){
  mtrain[is.na(mtrain[,i]), i] <- mean(mtrain[,i], na.rm = TRUE)
}

###Check
colSums(is.na(mtrain))

###Summary statistics
summary(mtrain)


###missing values test.
colSums(is.na(test))

###Impute with mean
for(i in 1:ncol(test)){
  test[is.na(test[,i]), i] <- mean(mtrain[,i], na.rm = TRUE)
}

###Check
colSums(is.na(test))

###Summary statistics
summary(test)

###descriptive statistics
summary(mtrain)

stargazer(mtrain, type = "text", title="summary statistics", digits=3)
```

```
###Seperate data by city
sj_train <- filter(mtrain,  city == 'sj')

iq_train <- filter(mtrain, city == 'iq')

###Seperate data by city
sj_test <- filter(test,  city == 'sj')

iq_test <- filter(test, city == 'iq')


###Check data
str(sj_train)

str(iq_train)

###Total dengue
par(mfrow = c(2, 2))
g1<-ggplot(sj_train, aes(x = total_cases)) +
  theme_bw() +
  ggtitle('Cases of dengue in San Juan') +
  geom_histogram(color = "black", fill="skyblue") +
  theme(plot.title = element_text(hjust = 0.5))
g2<-ggplot(iq_train, aes(x = total_cases)) +
  theme_bw() +
  ggtitle('Cases of dengue in Iquitos') +
  geom_histogram(color = "black", fill="skyblue") +
  theme(plot.title = element_text(hjust = 0.5))
multiplot(g1,g2, cols=2)

##time plot
sj_train %>%
  mutate(index = as.numeric(row.names(.))) %>%

ggplot(aes(index, total_cases)) +
  geom_line(colour = 'dodgerblue') +
  ggtitle("Total Cases in San Juan")

iq_train %>%
  mutate(index2 = as.numeric(row.names(.))) %>%

ggplot(aes(index2, total_cases)) +
  geom_line(colour = 'dodgerblue') +
  ggtitle("Total Cases in Iquitos")

###Drop week_start_date
```

```
sj_train <- dplyr::select(sj_train, -week_start_date)
iq_train <- dplyr::select(iq_train, -week_start_date)
mtrain <- dplyr::select(mtrain, -week_start_date)


###Check data
str(sj_train)

str(iq_train)

# Converting character columns into numbers
sj_train <- as.data.frame(apply(sj_train,2,as.numeric))
sj_train$city <- rep('sj', nrow(sj_train))
iq_train <- as.data.frame(apply(iq_train,2,as.numeric))
iq_train$city <- rep('iq', nrow(iq_train))

result <- list(sj_train, iq_train )

str(result)

###Correlation matrix
m_sj_train <- data.matrix(sj_train)
m_sj_train <- cor(x = m_sj_train[,3:24], use = 'complete.obs', method = 'pearson')

m_sj_train

m_iq_train <- data.matrix(iq_train)
m_iq_train <- cor(x = m_iq_train[,3:24], use = 'everything', method = 'pearson')

m_iq_train

###Correlation bar plot
cm_sj_train <- data.frame(m_sj_train)[2:21,]
cm_sj_train <- dplyr::select(cm_sj_train, total_cases)

cm_iq_train <- data.frame(m_iq_train)[2:21,]
cm_iq_train <- dplyr::select(cm_iq_train, total_cases)

ggplot(cm_sj_train, aes(x= reorder(rownames(cm_sj_train), -total_cases), y = total_cases)) +
  geom_bar(stat = 'identity',color = "black", fill="dodgerblue") +
  theme_bw() +
  ggtitle('Correlation of variables in San Juan') +
  ylab('Correlation') +
  xlab('Variables') +
  coord_flip()

ggplot(cm_iq_train, aes(x= reorder(rownames(cm_iq_train), -total_cases), y = total_cases)) +
  geom_bar(stat = 'identity', color = "black", fill="dodgerblue") +
  theme_bw() +
  ggtitle('Correlation of variables in Iquitos') +
  ylab('Correlation') +
  xlab('Variables') +
  coord_flip()

###Boxplots and qq plots
par(mfrow = c(2,4))
boxplot(sj_train$reanalysis_specific_humidity_g_per_kg,
        col = "deepskyblue1",main ="Specific_Humidity" , xlab = "")
boxplot(sj_train$reanalysis_dew_point_temp_k,
        col = "deepskyblue2",main ="Dew_Point_Temp_k", xlab = "")
boxplot(sj_train$station_min_temp_c,
        col = "deepskyblue3",main ="Min_Temp_c", xlab = "")
boxplot(sj_train$reanalysis_relative_humidity_percent,
        col = "deepskyblue4",main ="Relative_Humidity", xlab="")
qqnorm(sj_train$reanalysis_specific_humidity_g_per_kg,
       col = "deepskyblue1",main = "Specific_Humidity" , xlab = "")
qqline(sj_train$reanalysis_specific_humidity_g_per_kg,
       col = "black",main ="")
qqnorm(sj_train$reanalysis_dew_point_temp_k,
       col = "deepskyblue2",main = "Dew_Point_Temp_k", xlab = "")
qqline(sj_train$reanalysis_dew_point_temp_k,
       col = "black",main ="")
qqnorm(sj_train$station_min_temp_c,
       col = "deepskyblue3",main = "Min_Temp_c", xlab = "")
qqline(sj_train$station_min_temp_c,
       col = "black",main ="")
qqnorm(sj_train$reanalysis_relative_humidity_percent,
       col = "deepskyblue4",main ="Relative_Humidity", xlab = "")
qqline(sj_train$reanalysis_relative_humidity_percent,
       col = "black",main ="")

####Building Models
###Model 1linear regression Total with stepwise selection
lrt <- step(lm(formula = total_cases ~
                 year+
                 weekofyear +
                 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
                , data = mtrain),direction="both")

summary(lrt)

###Prediction of total_cases in the data set linear regression
prediction_lm <-  predict(lrt, newdata=test, type = 'response')
data_prediction_lm <- data.frame('city' = rep('sj', length(prediction_lm) ),
                       'total_cases' = prediction_lm,
                       'weekofyear' = test$weekofyear,
                       'year' = test$year )

submission_format$total_cases <- as.numeric(c(data_prediction_lm$total_cases))
submission_format$total_cases <- round(submission_format$total_cases, 0)
write.csv(submission_format,file = 'Model_1.csv', row.names = F)


###Mode2 negative binomials
nbr <- step(glm.nb(formula = total_cases ~
                year+
                weekofyear +
                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
          , data = mtrain),direction="both")

summary(nbr)

###Prediction of total_cases in the data set linear regression
prediction_br <-  predict(nbr, newdata=test, type = 'response')
data_prediction_br <- data.frame('city' = rep('sj', length(prediction_br) ),
                       'total_cases' = prediction_br,
                       'weekofyear' = test$weekofyear,
                       'year' = test$year )

submission_format$total_cases <- as.numeric(c(data_prediction_br$total_cases))
submission_format$total_cases <- round(submission_format$total_cases, 0)
write.csv(submission_format,file = 'Model_2.csv', row.names = F)

###Model 3 Neural net
fit  <- nnet(total_cases ~
            year+
            weekofyear +
            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
          , data=mtrain, repeats=25, size=3, decay=0.1,
          linout=TRUE)
```

```
###Prediction of total_cases in the data set linear regression
prediction_n <-  predict(fit, newdata=test)
data_prediction_n <- data.frame('city' = rep('sj', length(prediction_n) ),
                               'total_cases' = prediction_n,
                               'weekofyear' = test$weekofyear,
                               'year' = test$year )

submission_format$total_cases <- as.numeric(c(data_prediction_n$total_cases))
submission_format$total_cases <- round(submission_format$total_cases, 0)
write.csv(submission_format,file = 'Model_3.csv', row.names = F)

###Model 4 Arima time series
sjts=ts(sj_train$total_cases, frequency=52, start=c(1990,04,30))
iqts=ts(iq_train$total_cases, frequency=52, start=c(2000,07,01))


par(mfrow=c(1,2))
plot(sjts)
plot(iqts)

fit <- auto.arima(sjts)
fit_Arima=forecast(fit, h = 260)
plot(forecast(fit))

fit_iq <- auto.arima(iqts)
fit_Arima_iq=forecast(fit_iq, h = 156)
plot(forecast(fit_Arima_iq))

###Prediction of total_cases in the data set auto.arima
arima_sj <- data.frame(submission_format[1:260,-4], total_cases = round(fit_Arima$mean))
arima_iq <- data.frame(submission_format[261:416,-4], total_cases =round(fit_Arima_iq$mean))

arima_solution <- bind_rows(arima_sj,arima_iq)

write.csv(arima_solution, file = 'Model_4.csv', row.names = F)

###Random forest with specific predictors
sj_rf <- randomForest(total_cases ~
                           reanalysis_specific_humidity_g_per_kg +
                           reanalysis_dew_point_temp_k +
                           station_avg_temp_c +
                           station_min_temp_c
                         , data = sj_train)

print(sj_rf)

prediction_sjrf<- predict(object=sj_rf, sj_test)
```

```
###Random
iq_rf <- randomForest(total_cases ~
                    reanalysis_specific_humidity_g_per_kg +
                    reanalysis_dew_point_temp_k +
                    station_avg_temp_c +
                    station_min_temp_c
                  , data = iq_train)

print(iq_rf)

prediction_iqrf<- predict(object=iq_rf, iq_test)

###Prediction of total_cases in the data set auto.arima
rf_sj <- data.frame(submission_format[1:260,-4], total_cases = round(prediction_sjrf))
rf_iq <- data.frame(submission_format[261:416,-4], total_cases =round(prediction_iqrf))

rf_sol <- bind_rows(rf_sj,rf_iq)

write.csv(rf_sol, file = 'Model_5.csv', row.names = F)
```

References:

Davis, R. A., & Wu, R. (2009). A negative binomial model for time series of counts. *Biometrika, 96*(3), 735-749. doi:http://dx.doi.org.turing.library.northwestern.edu/10.1093/biomet/asp029.

Bergh, F. v. d., Holloway, J. P., Pienaar, M., Koen, R., Elphinstone, C. D., & Woodborne, S. (2008). A comparison of various modeling approaches applied to cholera case data*. *ORiON, 24*(1), 17-36. Retrieved from http://turing.library.northwestern.edu/login?url=https://search-proquest-com.turing.library.northwestern.edu/docview/200015189?accountid=12861

Wang, J., & Deng, Z. (2016). Modeling and prediction of oyster norovirus outbreaks along gulf of mexico coast. *Environmental Health Perspectives (Online), 124*(5), 627. Retrieved from http://turing.library.northwestern.edu/login?url=https://search-proquest-com.turing.library.northwestern.edu/docview/1806104356?accountid=12861

Sharma, V. C., Frankenfield, D., Gupta, A., & Singh, R. K. (2015). Ensemble approach for zoonotic disease forecasting using machine learning techniques. *International Journal of Business Analytics and Intelligence, 3*(2), 11-24. Retrieved from http://turing.library.northwestern.edu/login?url=https://search-proquest-com.turing.library.northwestern.edu/docview/1845229715?accountid=12861

Artola, C., Pinto, F., & de, P. G. (2015). Can internet searches forecast tourism inflows? *International Journal of Manpower, 36*(1), 103-116. Retrieved from http://turing.library.northwestern.edu/login?url=https://search-proquest-com.turing.library.northwestern.edu/docview/1663803663?accountid=12861