Dengue Epidemic Prediction with Regression Model

A One-on-One Comparison of R & Python

Md. Muminur Rahman

School of Computing & Mathematics
University of Derby

m.rahman23@unimail.derby.ac.uk
r.muminur@gmail.com

Abstract. The aim of the paper is to compare between two analytics environments in the context of developing a model for dengue fever outbreak prediction using climate, vegetation and dengue data of San Juan, Puerto Rico. For this purpose, R and Python as analytics tool and Multiple Linear Regression (MLR) as analytics technique have been chosen to develop the model. The capabilities, limitations and equation process of MLR is elaborated. Then a software selection framework has been developed and justified to compare between two analytics environments based on it. MLR model is developed using both of R and Python and prediction has been made with a considerable accuracy. Then a critical comparison between these two analytics environments has been made based on the justified analytics environment selection framework in the context of developing the prediction model. Finally, the report suggested the suitable tool in this context and recommended some future improvements.

Keywords: Big data, Dengue, Regression, Multiple Linear Regression, Predictive Model, Advanced Analytics, Machine Learning, R, Python

Table of Contents

1 Introduction

Dengue is one of the biggest seasonal epidemic in tropical countries. If the pattern of dengue epidemic is projected and predicted properly, it is possible to get preventive techniques. Many studies show that there is a strong relation between dengue and climate changes. For example, high temperature, precipitation and vegetation may influence the dengue epidemic. This paper aims to find relationships between climate and dengue epidemic and develop a prediction model for dengue epidemic. For this purpose, two analytics environment is chosen. Finally, the best environment for this purpose will be proposed in conclusion.

1.1 Problem Definition

There are specific weeks in a year when dengue outbreaks occur mostly. Predicting the weeks of peak incidences and maximum weekly incidence during the transmission season can help health officials target prevention messages and activities. It could also help hospital personnel make appropriate decisions about resource allocation (e.g. staffing) to ensure optimal patient care. On the other hand, there is a relationship between climate changes and dengue epidemic, as different studies showed. So, finding relationships between climate variables and dengue cases can help to build a predictive model for dengue outbreaks. Hence, a model needs to be developed by analysing the relationship between dengue cases and climate data to predict peak time of dengue outbreak and maximum weekly incidence.

1.2 Description of Data

There are many sources where climate and dengue data is available. But most of the dataset contains data about few years. But the dengue data of San Juan has been collected for a long time; from 1990 to 2013. Hence, this dataset has been selected for the project. For developing the predictive model, the datasets are collected from drivendata.com as they collected the climate data from four different sources and merged it with dengue data properly to make better predictive model. There are three datasets that have been used for this report:

- **Climate Data:** The climate data includes temperature, perception and vegetation data of Sun Juan gathered from various sources from 1990 to 2008. The climate dataset has been combined from four open datasets of US National Oceanic and Atmospheric Administration (Driven Data, 2017).
- **Dengue Data:** This includes total cases of dengue by week of the year. The dataset includes the dengue cases of San Juan from 1990 to 2008. The dataset is used for developing prediction model.
- **Test Data:** The dataset contains the dengue cases data with climate parameters from 2009 to 2013. The dataset is used as test data for checking the efficiency of the developed model.

1.3 Analytics Technique

Multiple Linear Regression (MLR) is used for solving problems stated in the section 1.1. Because the prediction is made on numeric continuous variables. Along with this, the problem also includes time series forecasting. Large number of successful applications have shown that MLR algorithms can be very useful tools for time-series modelling and forecasting (Ouahilal et al., 2016).

1.4 Analytics Environments

R and Python have been chosen as analytics environments for developing the model. Python is a popular an open-source, easy-to-use, portable, extensible, embeddable, high-level, interpreted, object-oriented programming language created by Guido Van Rossem in 1991, which emphasises code readability and productivity (Grover and Kar, 2017; Piatetsky, 2017). R is an open-source programming language and environment for statistical computing and graphics (The R Foundation, 2017). Both of these languages are popular now for machine learning and statistical modelling.

2 Critical Evaluation & Justification of Analytics Technique

Machine learning is "the training of a model from data that generalises a decision against a performance measure" (Gollapudi, 2016). In supervised machine learning, the system is trained by training dataset. The training dataset contains data and the correct output of the task based on that data. The system uses this data and "generates a function that maps inputs to desired outputs" (Ayodele, 2010). As the aim of the project is to forecast the dengue outbreaks based on past data, it is a supervised machine learning problem. There are many supervised learning algorithms which can do this task. However, Multiple Linear Regression (MLR) algorithm has been chosen for this scenario.

2.1 Capabilities of Multiple Linear Regression

MLR is a standard statistical technique which allows to answer questions that consider the role(s) that multiple independent variables play in a single dependent variable (Nathans, Oswald and Nimon, 2012). MLR can perform three types of job: (1) describing relationships among the dependent variables and the independent variables, (2) estimating the values of the dependent variables from the observed values of the independent variables, (3) identifying independent variables influencing on dependent variable (Schneider, Hommel and Blettner, 2010; Jeon, 2015).

2.2 How Multiple Linear Regression Works?

Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent

variable y (Zhao, 2012). So, there is only one dependent variable, which is dengue cases, and many independent variables. The equation of multiple linear regression is:

$$Y = a + b1*X1 + b2*X2 + ... + bp*Xp \quad (1)$$

Y is the dependant variable, which is the total-cases column in the dataset. a (Alpha) is the intercept. b variables denote the slope (Beta coefficient) for X variables. X variables denote the independent variables that are explaining the variance in Y. In our dataset, all climate variables are considered independent variables primarily. Then, insignificant independent variables is removed.

2.3 Reason for Choosing Multiple Linear Regression

The aim of the model to predict the dengue cases, which is continues variable, in peak season based on climate variables. MLR is a good choice when a continuous dependent variable is predicted, based on several independent variables (Gollapudi, 2016). Another aim of the model is to find the climate variables that influence on dengue outbreaks. One of the main purposes of MLR is to figure out independent variables influencing on dependant variable and to estimate the value of dependant variable according to the changes of independent variables (Jeon, 2015). In addition, MLR is easy to use and understand, fast and highly interpretable. It also doesn't require tuning of parameters like tuning the K parameter in K-Nearest Neighbours algorithm.

Many researchers used MLR model to predict dengue outbreak and influence of climate on dengue outbreak. Colón-González, Lake and Bentham (2011) developed a MLR model to look for associations between changes in the incidence rate of dengue fever and climate variability in the warm and humid region of Mexico. Anggraeni et al. (2017) also used MLR to predict number of dengue fever attack in Indonesia based on weather factors. The average MAPE of the model was less than 10%. Karim et al. (2012) developed a dengue prediction model for Dhaka city by analysing climate variables with a considerable accuracy.

2.4 Limitation of Multiple Linear Regression

Unlike decision trees, MLR cannot handle the missing values in dataset, which is a big limitation for this algorithm (Torgo, 2010). This problem can be handled through proper treatment of missing values. Sometimes, adding statistically non-significant variables into equation may increase the R^2. So, the model looks good theoretically but performs bad practically (Jeon, 2015). Proper selection of independent variables with appropriate technique can solve this kind of problem.

2.5 Analytics Environment Selection Framework

It is a puzzling problem to select one environment from the plethora of analytics environment, as there are lots of analytics environment with different capabilities and limitations. It is advised, the selection of a analytics environment should be approached in

the same way as the selection of any IT product by assessing the current and future state of use in the business and developing a set of selection criteria (Leventhal, 2010). Hence, a framework has been developed and justified. The environments will be compared against the criteria set in this framework.

Table 1. Analytics environment selection framework.

Requirements	Why it matters?
Is the system integration is easily possible with this environment?	Sometimes, a prediction model needs to be integrated into a production level application. Keeping integration options open helps ensure that the changes in user needs can be addressed properly (Pentaho, 2015; Parenteau, 2016).
How much faster the environment in processing tasks?	The processing speed of analytics environments are different. So, the processing speed for intended job must be assessed.
How much good and flexible is the visual capability of the tool?	Visualisation plays a vital role for explanatory analysis of data and helps finding streams and correlations of variables.
Is it easy to install, configure and manage the platform?	Some platforms are complex to set and install. So, the installation complexity of the platform needs to be taken into account (Pentaho, 2015).
Are adequate libraries and packages available for the environment?	Packages and libraries make data cleansing, manipulation, aggregation and visualisation process easier and save time. The availability of enough packages must be taken into account while selecting analytics tool.
Is learning curves of the platform easy and is community support good?	The easiness of learning curves and adequate support of community make customised solution development faster and easier.

3 Critical Evaluation of Analytics Environments

Python and R are used to perform prediction for this project. Both have some capabilities and limitations in the context of the project and evaluation framework. In following lines, an evaluation between Python and R is given in the context of developing the model.

- **Setting Development Environment:** Setting environment for R is so easy. Only the correct exe file needs to be downloaded from the official website and installed in PC. RStudio is a powerful integrated development environment (IDE) that is mostly used for R. In contrast, setting data science development environment for Python is tricky as it has many different distributions for data science which contain different data science libraries. Beginners get confused while choosing desired distribution which can serve his/her purpose. Along with this, there is not any powerful and free

IDE like RStudio for Python data science. To overcome this, Jupyter Notebook, an open-source web application, has been used in this project for Python.

- **Data Preparation & Cleansing:** The training dataset contains 209 rows that have missing values. The north-east vegetation and north-west vegetation columns have almost 150 missing values. So, these columns have been removed from training and testing dataset as shown in Snippet 1 in Appendix 1. The other missing values has been imputed with mean value of the column. There are Python and R libraries and functions for missing value imputation. So, this task was handled easily in both of the environments. As MLR can be implemented only numeric columns in R, four columns in the dataset have been converted to numeric class using `as.numeric` function, because these columns belong to factor and integer class. In Python, these columns are considered as integer. So, there is no need to change the type of the column datatype.
- **Exploratory Analysis & Visualisation:** The exploratory analysis has been done using both of Python and R. The analysis illustrates that the dengue season is from week 38 to week 48, which starts from mid-September and ends on November (See Appendix 2: Fig 5). Correlation matrix is generated to find correlation between dengue cases and climate variables. The matrix illustrates that there is not any strong correlation between dengue cases and any specific parameter. Instead, dengue cases correlate slightly with humidity, air temperature and vegetation (See Appendix 2: Fig 2). Fig 6 in appendix 2 illustrates that the dengue outbreak comes continuously after few years. But the number of cases in dengue outbreak decreases. On the other hand, the most cases of dengue occurs in week number 40. There are some libraries (e.g. R commander, Rattle, GrapheR etc.) which enables user to conduct basic exploratory analysis, graph generation and model development easily using GUI without coding. R commander and Rattle is used for basic exploratory analysis with R. But there is not any GUI library for Python (Ohri, 2012). Along with this, the main purpose of R was statistical analysis and visualisation. So, R comes with many sophisticated libraries for lucrative data visualisation which help developers generate a specific plot in dozens of ways easily. On the other hand, Python lacks those libraries. Most of the time, there is no way without using Matplotlib and seaborn library (Castle, 2017).
- **Model Development & Validation:** The first model in Python made with all of the variables. After getting the result of this model, another model is made with the variables that have a p-value under 0.5. The adjusted R^2 of the model is 0.069, which is very poor. There is not any p-value shown in Python for the model. After that, prediction has been made with test data, where the mean squared error was 66.40.
  For choosing best model for prediction, stepwise backward elimination process has been chosen in R. The variables used to develop model in Python are used to develop model in R. The model gives a result with p-value 2.2e-16 which shows that the result is significant. The p-value is satisfactory as it is less than 0.5 which indicates that the null hypothesis is acceptable. On the other hand, the adjusted R^2 is 0.1064, which is much satisfactory than the model developed with Python. After that, the predicted values have been tested with test data. The mean squared error of the prediction was 41.87 which means that the prediction of the model developed with R is

closer to the actual values than Python. Appendix 2: Fig 4 illustrates the model. The QQ plot shows that the residuals are mostly lined well on the straight dashed line which means the prediction with R is considerably good.

There is not any automatic model selection method in Python. So, the model has been developed and chosen manually. On the other hand, there are libraries in R to select best fit models using different techniques which makes model selection process easier. Along with this, the easiest way to determine accuracy of a linear model in Python is finding R^2. Other methods are a little bit complex for beginners. In contrast, the default `summary` function of R is enough to evaluate a linear model.

- **Processing Speed:** Python is so much faster than R while processing data. A study shows that R has half processing speed compared to Python. For example, elapsed times for doing the Metropolis algorithm computations of N=15000 with R and Python are 0.243 and 0.08 second respectively (Kouatchou, 2017).
- **System Integration:** R was invented for statistical analysis only. So it lacks common development tools. On the other hand, Python is a popular general purpose language. So, while developing data science based application, Python is the best choice. If the project goal is to analyse data statistically then R is the better choice, as Python lacks many statistical functions (Paruchuri, 2015).
- **Learning Curve & Community Support:** R is a little bit complex for new learners, while Python is popular for easy learning curves. But in the context of statistical analysis, R community is bigger and more powerful than Python. The Python community for general purpose language is good, but the Python community for data science is growing (Willems, 2015).

4 Conclusion & Recommendation

The predictive model for dengue cases has been developed using R and Python. A comparison also made between these two environments in the context of the project. The evaluation of environments illustrated that R is most powerful tool that makes statistical analysis, data visualisation & manipulation and model development & evaluation easier and faster, while Python lacks many built-in data analysis features which makes statistical analysis and model development quiet harder in Python. But as a multipurpose language, Python is the best when it comes to develop predictive models for applications and production server where high processing speed and ease of integration are anticipated. So, R fits best with the context of this project and enables rapid model development and prediction. Along with this, the model developed with R for this project gives better result than the Python model.

If this model is used in part of an application to visualise climate and dengue cases data with real time prediction and visualisation for dengue epidemic, then Python will get priority. Because the application needs to collect data from different sources in real time which need integration with different production systems and faster data process. R lacks many common process for system integration and has a slower processing speed. So, Python should be used for that scenario to get the best performance.

5 Bibliography

1. Anggraeni, W. *et al.* (2017) 'Modified Regression Approach for Predicting Number of Dengue Fever Incidents in Malang Indonesia', *Procedia Computer Science*. (4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia), 124, pp. 142–150. doi: 10.1016/j.procs.2017.12.140.
2. Ayodele, T. O. (2010) 'Types of machine learning algorithms', in *New advances in machine learning*. InTech.
3. Castle, N. (2017) 'R vs. Python: What Language is Best for Building Data Models?', *DataScience.com*, 20 July. Available at: https://www.datascience.com/blog/r-vs-python-for-data-models-data-science (Accessed: 8 December 2017).
4. Colón-González, F. J., Lake, I. R. and Bentham, G. (2011) 'Climate Variability and Dengue Fever in Warm and Humid Mexico', *The American Journal of Tropical Medicine and Hygiene*, 84(5), pp. 757–763. doi: 10.4269/ajtmh.2011.10-0609.
5. Driven Data (2017) *DengAI: Predicting Disease Spread*, *Driven Data*. Available at: https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/page/82/ (Accessed: 9 December 2017).
6. Gollapudi, S. (2016) *Practical Machine Learning*. Packt Publishing.
7. Grover, P. and Kar, A. K. (2017) 'Big Data Analytics: A Review on Theoretical Contributions and Tools Used in Literature', *Global Journal of Flexible Systems Management*, 18(3), pp. 203–229. doi: 10.1007/s40171-017-0159-3.
8. Jeon, J. (2015) 'The Strengths and Limitations of the Statistical Modeling of Complex Social Phenomenon: Focusing on SEM, Path Analysis, or Multiple Regression Models', *Int. J. Soc. Behav. Educ. Econ. Bus. Ind. Eng*, 9, pp. 1559–1567.
9. Karim, M. N. *et al.* (2012) 'Climatic factors influencing dengue cases in Dhaka city: A model for dengue prediction', *The Indian Journal of Medical Research*, 136(1), pp. 32–39.
10. Kouatchou, J. (2017) *Basic Comparison of Python, Julia, R, Matlab and IDL*, *NASA Modeling Guru*. Available at: https://modelingguru.nasa.gov/docs/DOC-2625 (Accessed: 8 December 2017).
11. Leventhal, B. (2010) 'An introduction to data mining and other techniques for advanced analytics', *Journal of Direct, Data and Digital Marketing Practice*, 12(2), pp. 137–153. doi: 10.1057/dddmp.2010.35.
12. Nathans, L. L., Oswald, F. L. and Nimon, K. (2012) 'Interpreting multiple linear regression: A guidebook of variable importance', *Practical Assessment, Research & Evaluation*, 17(9).
13. Ohri, A. (2012) *R for Business Analytics*. Springer Science & Business Media.
14. Ouahilal, M. *et al.* (2016) 'A comparative study of predictive algorithms for business analytics and decision support systems: Finance as a case study', in *2016 International Conference on Information Technology for Organizations Development (IT4OD). 2016 International Conference on Information Technology for Organizations Development (IT4OD)*, pp. 1–6. doi: 10.1109/IT4OD.2016.7479258.
15. Parenteau, J. (2016) *Evaluation Guide: How to choose the right modern BI & analytics platform*. Tableau Software. Available at: https://www.tableau.com/learn/whitepapers/evaluation-guide-how-choose-right-modern-bi-analytics-platform (Accessed: 15 December 2017).
16. Paruchuri, V. (2015) 'Python vs R: head to head data analysis', *Dataquest*, 7 October. Available at: http://www.dataquest.io/blog/python-vs-r/ (Accessed: 5 January 2018).
17. Pentaho (2015) 'Embedded Analytics Vendor Selection Guide'. Pentaho Corporation. Available at: http://www.pentaho.com/sites/default/files/uploads/resources/pentaho-oem-embed-eval-guide.pdf (Accessed: 6 January 2018).

18. Piatetsky, G. (2017) *Python vs R – Who Is Really Ahead in Data Science, Machine Learning?*, *KDnuggets*. Available at: https://www.kdnuggets.com/2017/09/python-vs-r-data-science-machine-learning.html (Accessed: 8 December 2017).
19. Schneider, A., Hommel, G. and Blettner, M. (2010) 'Linear regression analysis: part 14 of a series on evaluation of scientific publications', *Deutsches Ärzteblatt International*, 107(44), p. 776.
20. Torgo, L. (2010) *Data Mining with R: Learning with Case Studies*. Taylor & Francis.
21. Willems, K. (2015) *Choosing R or Python for data analysis? An infographic*, *DataCamp*. Available at: http://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis (Accessed: 8 December 2017).
22. Zhao, Y. (2012) *R and data mining: Examples and case studies*. Academic Press.

6 Appendix 1 : Code Snippets

```
#checking na rows
nrow(data[!complete.cases(data),])

#Handling NA in train data
data[22] <- lapply(data[22], as.numeric)
sapply(data, function(x) sum(is.na(x)))
sum(is.na(data))
mean(is.na(data))
colMeans(is.na(data))
c_data <- na.omit(data)

#Handling NA in test data
test <- test[, -c(4,5)]
test[22] <- lapply(test[22], as.numeric)
sapply(test, function(x) sum(is.na(x)))
sum(is.na(test))
mean(is.na(test))
colMeans(is.na(test))
c_test <- na.omit(test)
```

Snippet 1: Handling missing values in training and testing data with R.

```
#Plotting
#Histogram of week of year

require(plotly)
library(plotly)
barplot_of_dengue <- plot_ly(
x = data$weekofyear,
y = data$total_cases,
name = "Barchart",
type = "bar"
) %>%
layout(title = "Dengue cases by week of year")

# Basic line plot with points
library(plotly)

p <- plot_ly(data, x = data$year, y = data$total_cases, type = 'scatter', mode =
'lines')
```

Snippet 2: Plotting with R.

```
#first linear model

model1 <- lm(total_cases ~ ndvi_se + ndvi_sw + precipitation_amt_mm + rea-
nalysis_air_temp_k + reanalysis_avg_temp_k + reanalysis_dew_point_temp_k +
reanalysis_max_air_temp_k + reanalysis_min_air_temp_k + reanaly-
sis_precip_amt_kg_per_m2 + reanalysis_relative_humidity_percent + reanaly-
sis_sat_precip_amt_mm + reanalysis_specific_humidity_g_per_kg + reanaly-
sis_tdtr_k +
station_avg_temp_c + station_diur_temp_rng_c +
station_max_temp_c + station_min_temp_c + station_precip_mm,  data =
c_data)
#second linear model
model2 <- lm(total_cases ~ reanalysis_max_air_temp_k +
reanalysis_specific_humidity_g_per_kg + station_diur_temp_rng_c + sta-
tion_max_temp_c, data = c_data)

#finale linear model by automatic backward elimination
library(MASS)
step <- stepAIC(model1, direction="backward", trace = "FALSE")
step$anova

model3 <- lm(total_cases ~ precipitation_amt_mm + reanalysis_avg_temp_k +
reanalysis_dew_point_temp_k + reanalysis_max_air_temp_k + reanaly-
sis_precip_amt_kg_per_m2 + reanalysis_specific_humidity_g_per_kg + reanaly-
sis_tdtr_k + station_diur_temp_rng_c + station_max_temp_c, data = c_data)

anova(model1, model2, model3)
par(mfrow = c(2,2))
plot(model3)
```

Snippet 3: Model development and best model selection with R.

```
prediction <- predict(model3, c_test)
c_test$prediction <- prediction
actuals_preds <- data.frame(cbind(actuals=c_test$total_cases, predicteds=pre-
diction))
correlation_accuracy <- cor(actuals_preds)
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actu-
als_preds, 1, max))
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actu-
als_preds$actuals)
library(Metrics)
mse(actuals_preds$actuals, actuals_preds$predicteds)
```

Snippet 4: Prediction with R.

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
import seaborn as sns
import statsmodels.api as sm
from sklearn import linear_model
import numpy as np
from IPython.display import HTML, display
from statsmodels.formula.api import ols
import matplotlib.pyplot as plt

# For statistics. Requires statsmodels 5.0 or more
from statsmodels.formula.api import ols
# Analysis of Variance (ANOVA) on linear models
from statsmodels.stats.anova import anova_lm
data = pd.read_csv("F:/Google Drive/UoD/Advanced Analytics/Data
Sets/cleandata.csv")
test = pd.read_csv("F:/Google Drive/UoD/Advanced Analytics/Data Sets/den-
gue_features_test.csv")
data.isnull().sum()
```

Snippet 5: Library integration & data reading iin Python.

```
for i in set(data['year']):
   df = data[data['year'] == i]
   df.set_index('weekofyear', drop = True, inplace = True)
   plt.plot(df['total_cases'], alpha = .3)

data.groupby('weekofyear')['total_cases'].mean().plot(c = 'k', figsize = (10,4))
plt.legend(set(data['year']), loc='center left', bbox_to_anchor=(1, .5))

plt.title('Dengue Cases by Week of the Year')
plt.xlabel('Week of the Year')
plt.ylabel('Number of Cases')
```

Snippet 6: Plotting with Python.

```
import statsmodels.formula.api as smf
model = smf.ols("""total_cases ~ 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 + reanaly-
sis_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= data).fit()

# print the coefficients
model.params

# print a summary of the fitted model
model.summary()

import statsmodels.formula.api as smf
model2 = smf.ols("""total_cases ~ ndvi_ne + ndvi_nw +
                  precipitation_amt_mm +
                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_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= data).fit()

# print the coefficients
model2.params

# print a summary of the fitted model
model2.summary()

feature_cols = ['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', 'rea-
nalysis_min_air_temp_k', 'reanalysis_precip_amt_kg_per_m2', 'reanalysis_rela-
tive_humidity_percent', 'reanalysis_sat_precip_amt_mm', 'reanalysis_spe-
cific_humidity_g_per_kg', 'reanalysis_tdtr_k',
                 'station_avg_temp_c', 'station_diur_temp_rng_c', 'sta-
tion_max_temp_c', 'station_min_temp_c', 'station_precip_mm']
x = data[feature_cols]
y = data.total_cases

# follow the usual sklearn pattern: import, instantiate, fit
```

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
model = lm.fit(x, y)

# print intercept and coefficients
print (lm.intercept_)
print (lm.coef_)
```

Snippet 7: Model creation in Python.

```
test_data = test[feature_cols]
predictions = lm.predict(test_data)
# RMSE
from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(test['total_cases'], predictions)))
## The line / model
import matplotlib.pyplot as plt
plt.scatter(test['total_cases'], predictions)
```

Snippet 8: Prediction in Python.

7 Appendix 2 : Figures

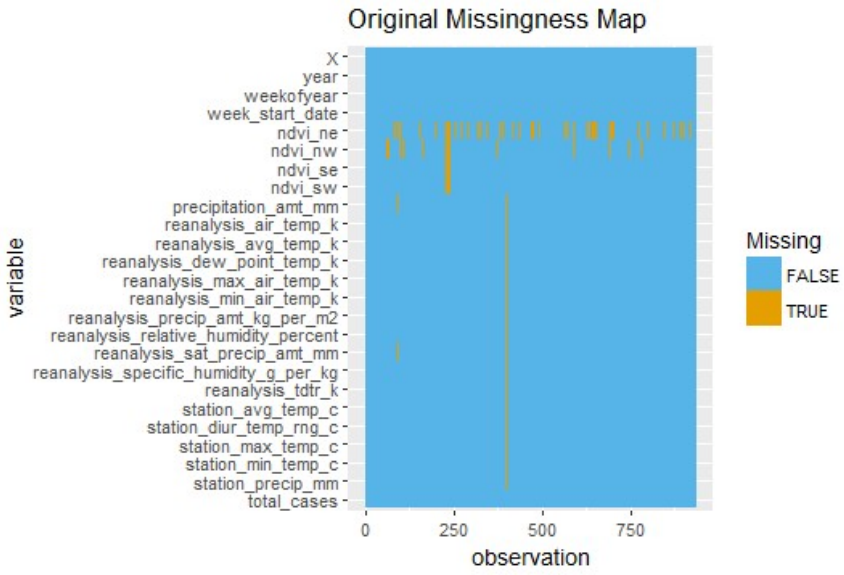

Fig. 1. Missing value mapping with R.

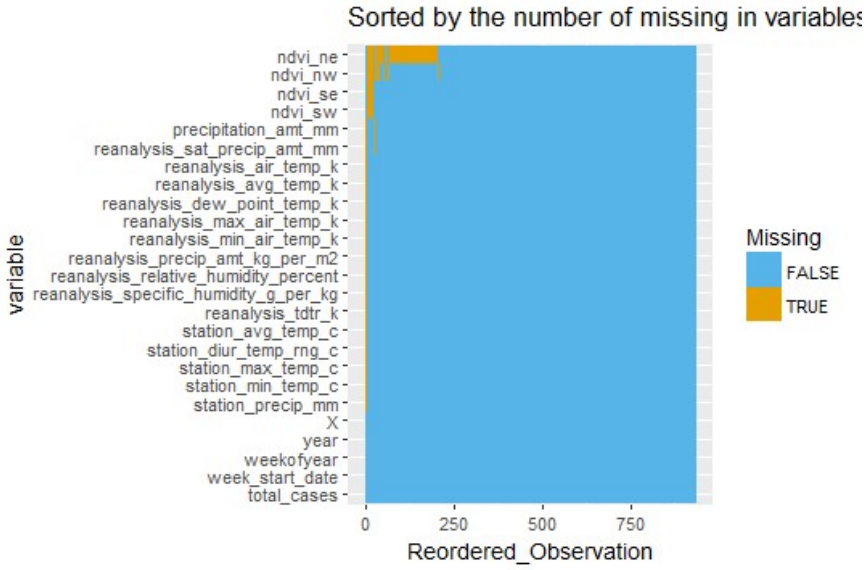

Fig. 2. Missing values in sorted way with R.

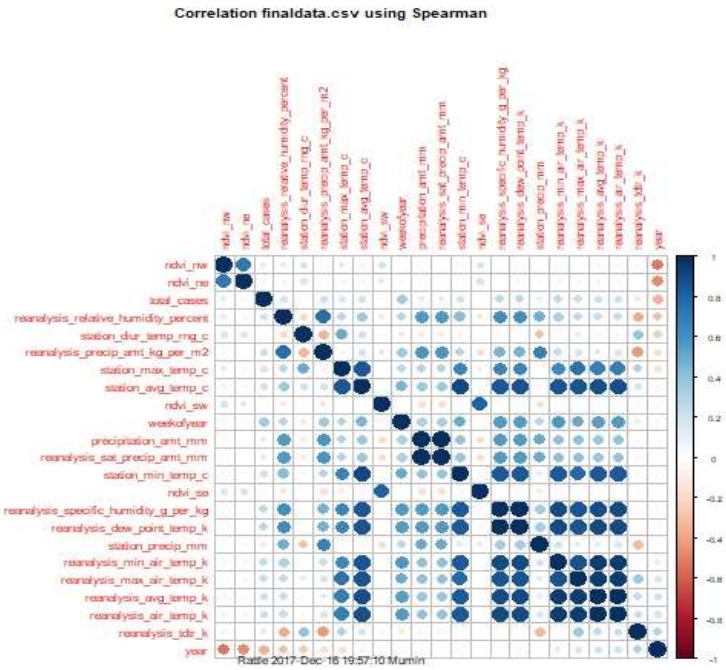

Fig. 3. Correlation matrix with R.

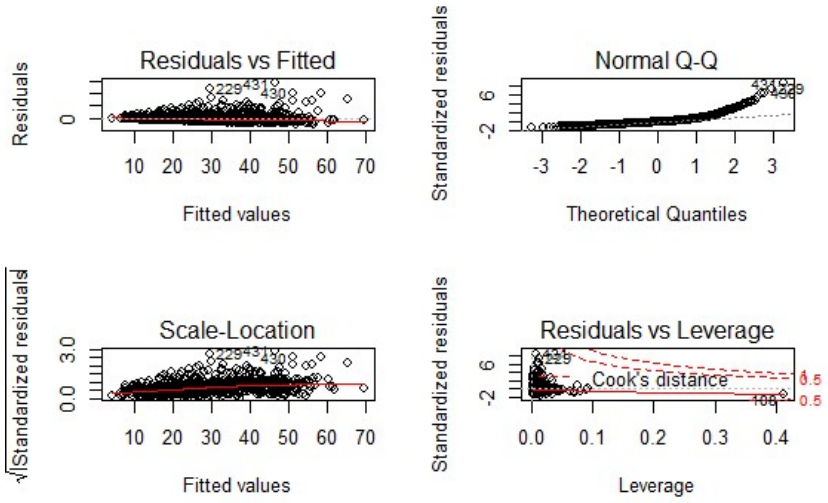

Fig. 4. Plots of prediction model using R.

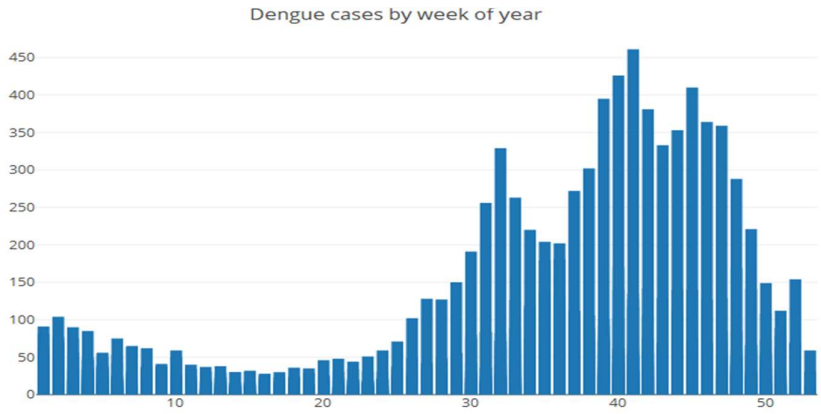

Fig. 5. Dengue cases by week of year with R.

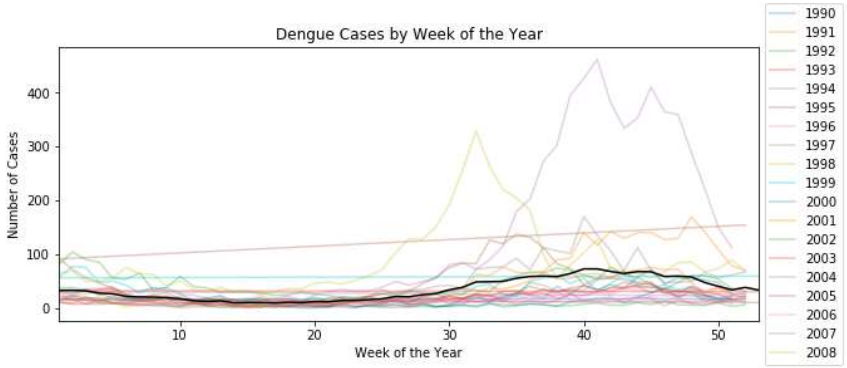

Fig. 6. Number of dengue cases over the weeks by years with Python.

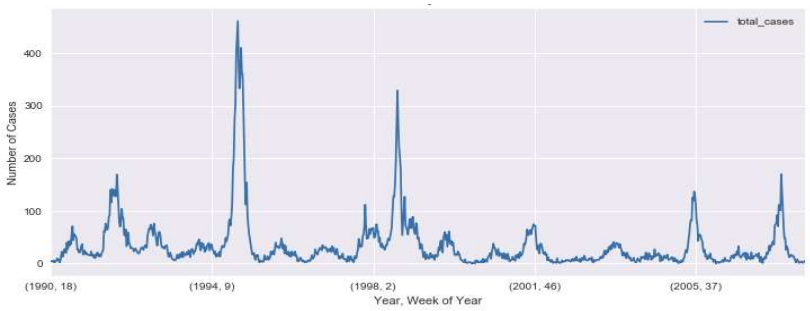

Fig. 7. Dengue outbreaks over the years by Python.

```
> summary(model3)

Call:
lm(formula = total_cases ~ precipitation_amt_mm + reanalysis_avg_temp_k +
    reanalysis_dew_point_temp_k + reanalysis_max_air_temp_k +
    reanalysis_precip_amt_kg_per_m2 + reanalysis_specific_humidity_g_per_kg +
    reanalysis_tdtr_k + station_diur_temp_rng_c + station_max_temp_c,
    data = c_data)

Residuals:
    Min      1Q  Median      3Q     Max
-53.473 -18.090  -7.586   6.173 282.765

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                            1.295e+04  3.933e+03   3.293 0.001029 **
precipitation_amt_mm                  -5.320e-02  3.068e-02  -1.734 0.083302 .
reanalysis_avg_temp_k                 -5.397e+00  3.646e+00  -1.480 0.139158
reanalysis_dew_point_temp_k           -5.350e+01  1.354e+01  -3.953 8.33e-05 ***
reanalysis_max_air_temp_k              1.202e+01  3.474e+00   3.460 0.000566 ***
reanalysis_precip_amt_kg_per_m2        6.517e-02  4.210e-02   1.548 0.121938
reanalysis_specific_humidity_g_per_kg  5.793e+01  1.401e+01   4.136 3.86e-05 ***
reanalysis_tdtr_k                     -1.115e+01  3.281e+00  -3.399 0.000705 ***
station_diur_temp_rng_c                4.845e+00  1.941e+00   2.497 0.012717 *
station_max_temp_c                    -3.267e+00  1.393e+00  -2.346 0.019177 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33.17 on 901 degrees of freedom
Multiple R-squared:  0.1152,    Adjusted R-squared:  0.1064
F-statistic: 13.04 on 9 and 901 DF,  p-value: < 2.2e-16
```

Fig. 8. Summary of the model with R.

OLS Regression Results

Dep. Variable:	total_cases	**R-squared:**	0.078
Model:	OLS	**Adj. R-squared:**	0.069
Method:	Least Squares	**F-statistic:**	8.756
Date:	Sun, 07 Jan 2018	**Prob (F-statistic):**	1.04e-12
Time:	02:47:19	**Log-Likelihood:**	-4976.6
No. Observations:	936	**AIC:**	9973.
Df Residuals:	926	**BIC:**	1.002e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	9284.3600	5826.452	1.593	0.111	-2150.222	2.07e+04
precipitation_amt_mm	-0.0734	0.045	-1.617	0.106	-0.163	0.016
reanalysis_avg_temp_k	-10.5325	5.366	-1.963	0.050	-21.064	-0.001
reanalysis_dew_point_temp_k	-40.8821	20.083	-2.036	0.042	-80.296	-1.468
reanalysis_max_air_temp_k	17.4138	5.093	3.419	0.001	7.418	27.410
reanalysis_precip_amt_kg_per_m2	0.0623	0.063	0.995	0.320	-0.061	0.185
reanalysis_specific_humidity_g_per_kg	42.5101	20.757	2.048	0.041	1.775	83.246
reanalysis_tdtr_k	-21.0138	4.845	-4.337	0.000	-30.522	-11.506
station_diur_temp_rng_c	3.7315	2.859	1.305	0.192	-1.879	9.342
station_max_temp_c	1.3572	2.028	0.669	0.503	-2.622	5.337

Omnibus:	868.576	**Durbin-Watson:**	0.126
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	26929.846
Skew:	4.328	**Prob(JB):**	0.00
Kurtosis:	27.811	**Cond. No.**	1.87e+06

Fig. 9. Summary of the MLR model with Python.