

# Rainfall Forecasting : A Regression Case Study

Ebenezer Ajay Williams<sup>1</sup>, Aditya Vazhipokkil Manoharan<sup>2</sup>, Arvind Mohan<sup>3</sup>

<sup>1</sup>B.Tech IT, SSN College of Engineering

<sup>2,3</sup>B.E CSE, SSN College of Engineering

\*\*\*

**Abstract:** The present paper analyses the monthly rainfall data of various regions of India between 1905-2015. Multiple linear regression is used to predict the average rainfall using the previous months' data from the corresponding time period. Prediction is done month wise according to the subdivision and compared with the actual value

**Key words:** India, rainfall, prediction, multiple linear regression ,residuals

## 1. Introduction:

India is an agricultural country and the success or failure of the harvest and water scarcity in any year are always considered with the greatest concern. Rainwater is the primary source of water for the citizens of the country with glaciers serving as a secondary source. Rainfall in India can be a result of several factors ranging from depressions in the Bay of Bengal to the mighty monsoon. Thus for a country that is so dependant on rainfall, predicting the rainfall could help the Government as well as the people to formulate plans for the upcoming years and be economical with the use of their current resources.

## 2.Literature Survey:

Guhathakurta (2006) postulates an Artificial Neural Network method in predicting monsoon-rainfall in the state of Kerala, This paper proved that a neural network approach could be applicable to predict rainfall over districts of Kerala up to 2003. But, a major drawback of this paper is that it did not analyze the autocorrelation structure of the rainfall time series and chose the input matrix quite arbitrarily. Hasternrath (1988) discussed the usefulness of regression model in predicting the rainfall during the monsoon season.

## 3.Dataset:

The dataset was shared by the Indian Government, and contains rainfall data for 36 meteorological sub-divisions in india and is recorded monthwise and scaled in mms. The dataset contains information pertaining to the time period 1901 - 2015.

Source: <https://data.gov.in/catalog/rainfall-india>

## 3.1.Missing data analysis and imputation :

The dataset contains certain missing values. Most were certain odd months in a year. But in certain cases, data were missing for an entire year. This could possibly have been due to lost data/faulty measuring. The following table visualises the missing data.

	Y	F	M	J	A	A	J	M	S	J	O	D	N
	E	A	A	P	U	U	A	E	U	C	E	O	
	R	B	Y	N	R	G	N	R	P	L	T	C	V
4090	1	1	1	1	1	1	1	1	1	1	1	1	0
2	1	1	1	0	1	1	1	1	1	1	1	1	1
2	1	0	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	0	1	1	1	1	1
2	1	1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	0	1
2	1	1	1	1	1	1	1	1	1	1	1	0	0
1	1	0	1	0	1	1	1	0	1	1	1	1	3
1	1	1	1	1	1	1	1	1	1	0	1	0	0
2	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	1	0	0	0	0	0
1	1	1	0	1	1	0	0	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0
												1	1
	0	3	3	4	4	4	5	6	6	7	7	0	1
													70

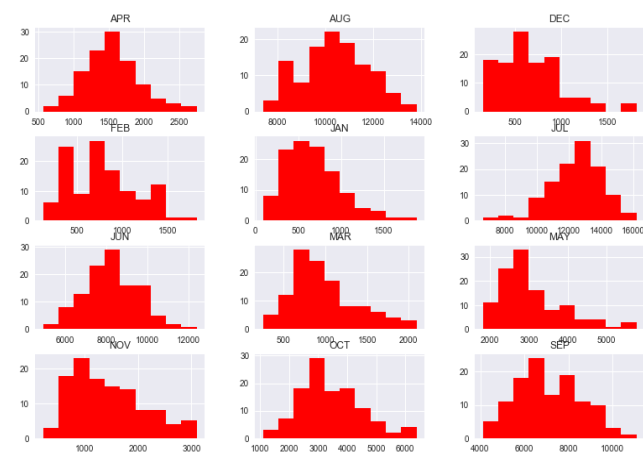
The 0s indicate missing values and the left column indicates number of rows with such missing pattern. As seen above (bottom right), the total number of missing

entries is 70. Since this is a small number compared to the total number of entries - 49392 (~0.41%), we decide to impute the missing data based on the rest of it.

Since there is an adequate amount of data to gather the information, we use predictive mean matching to impute the data. This uses chained equations so that the predicted values are more like real since they are drawn directly from the existing data.

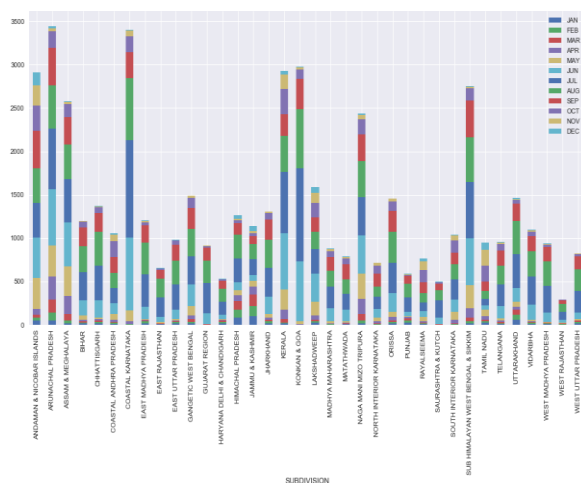
**4.Initial Analysis :**

Analysis of the data is done by visualizing the data through a series of plots to gain knowledge of the data.



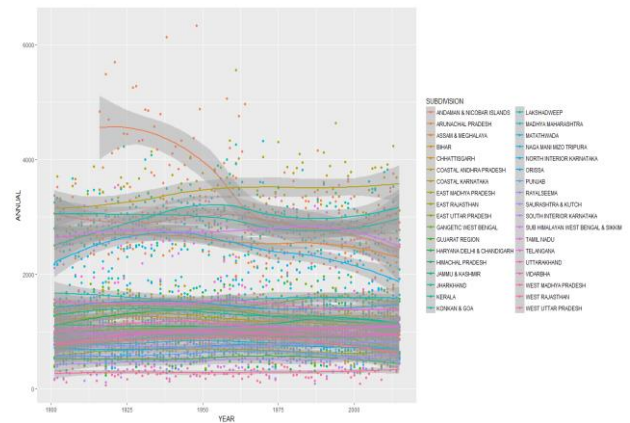
On analysing month wise, we find that the highest amount of rainfall is in July and May, and the lowest in December.

**Statewise:**



As seen above, Karnataka and Arunachal get the most amount of rainfall throughout the year, followed by Kerala, Goa and Bengal.

**Scatter plot:**



(Scatter plot:1)

The first thing noticeable from the plot above is how the areas with low rainfall are visibly separated from the areas with high rainfall visibly the ones with low rainfall range from 0-2000 and the ones above 2000 are high. As evident above, the annual rainfall over the years for most states has remained the same for the most part. However, in Arunachal Pradesh, the rainfall was remarkably high from 1925 to 1950 and then went down in the succeeding years. (Orange line)

Similarly, in Orissa, it rose, peaking around 1940, and then fell again ( Blue line).Additionally Konkan and Goa have experienced a variable range of rainfall throughout the period.

**5.Rainfall Prediction using Regression :**

Given the dataset , we can predict the rainfall month wise for every year. The prediction is done for the months June to December. As there are 36 different subdivisions there may be 36 different models for prediction.Since there may be more than one independent variable ,we may use multiple linear regression.There may be six columns of which the first five represent independent variables and the sixth represents the dependant variable. For every month predicted ,the values may be appended on to the dataset . There may be a total of eight iterations to predict the rainfall from May to December. The Mean Absolute Difference is calculated as the Absolute difference between the predicted values and the actual values.

The difference between the observed value of the dependent variable ( $y$ ) and the predicted value ( $\hat{y}$ ) is called the **residual** ( $e$ ). Each data point has one residual. Residual = Observed value - Predicted value

$$e = y - \hat{y}$$

In other words the distance between the regression function line and the plotted data would be termed as the residual.

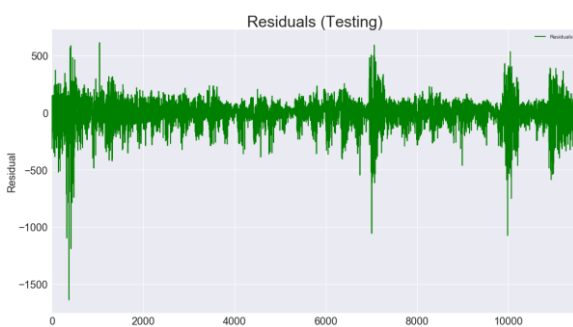
The Mean absolute difference serves as the residual in this case. During statistical analysis it is important to evaluate how well the model fits the data and that the data meet the assumptions of the model.

The dataset is split up into a training set and testing set with 60 percent randomly chosen for training and 40 percent randomly chosen for testing. This split is done so in order to perform a classification of high, medium or low rainfall. However this paper does not include classification which is a topic of further research.

### 6.Results

Mean Absolute Difference in Training set: 75.5054636791

Mean Absolute Difference in Testing set: 76.8001644151



### 7.Python Code for Regression :

```
subdivs = df['SUBDIVISION'].unique()
num_of_subdivs = subdivs.size

df_res_training = pd.DataFrame(columns=np.array(['Residuals']))
df_res_testing = pd.DataFrame(columns=np.array(['Residuals']))
list_mad_training = []
mean_abs_diff_training = 0
list_mad_testing = []
```

```
mean_abs_diff_testing = 0
for subdiv in subdivs:
    df1 = df[df['SUBDIVISION']==subdiv]
    df2 = df1[[months[0],months[1],months[2],months[3],months[4]]]
    df2.columns = np.array(['x1','x2','x3','x4','x5'])
    for k in range(1,8):
        df3 = df1[[months[k],months[k+1],months[k+2],months[k+3],months[k+4]]]
        df3.columns = np.array(['x1','x2','x3','x4','x5'])
        df2 = df2.append(df3)
    df2.index = range(df2.shape[0])
    msk = np.random.rand(len(df2)) < 0.6
    df_train = df2[msk]
    df_test = df2[~msk]
    df_train.index = range(df_train.shape[0])
    df_test.index = range(df_test.shape[0])
    reg = linear_model.LinearRegression()
    reg.fit(df_train.drop('x5',axis=1),df_train['x5'])
    predicted_values = reg.predict(df_train.drop('x5',axis=1))
    residuals = predicted_values-df_train['x5'].values
    df_res_training.append(pd.DataFrame(residuals,columns=np.array(['Residuals'])))
    mean_abs_diff_training = mean_abs_diff_training + np.sum(np.abs(residuals))
    list_mad_training.append(np.mean(np.abs(residuals)))

    predicted_values = reg.predict(df_test.drop('x5',axis=1))
    residuals = predicted_values-df_test['x5'].values
    df_res_testing.append(pd.DataFrame(residuals,columns=np.array(['Residuals'])))
    mean_abs_diff_testing = mean_abs_diff_testing + np.sum(np.abs(residuals))
    list_mad_testing.append(np.mean(np.abs(residuals)))
df_res_training.index = range(df_res_training.shape[0])
mean_abs_diff_training = mean_abs_diff_training/df_res_training.shape[0]
print('Overall (Training): ' + str(mean_abs_diff_training))

fig = plt.figure(figsize=(18,10))
ax = fig.add_subplot(111)
df_res_training.plot(line='c',ax=ax,fontsize=20)
#ax.xaxis.set_ticklabels([])
plt.ylabel('Residual')
ax.title.set_fontsize(30)
ax.xaxis.label.set_fontsize(20)
ax.yaxis.label.set_fontsize(20)

df_res_testing.index = range(df_res_testing.shape[0])
mean_abs_diff_testing = mean_abs_diff_testing/df_res_testing.shape[0]
```

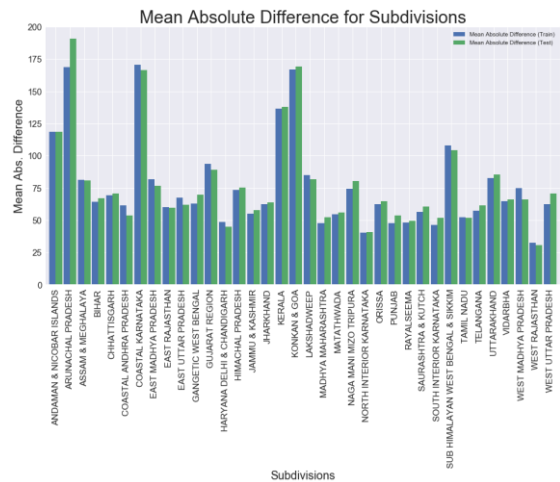
```

print('Overall MAD (Testing): ' +
str(mean_abs_diff_testing))
fig = plt.figure(figsize=(18,10))
ax = fig.add_subplot(111)
df_res_testing.plot.line(title='Residuals (Testing)',
color='m',ax=ax,fontsize=20)
#ax.xaxis.set_ticklabels([])
plt.ylabel('Residual')
ax.title.set_fontsize(30)
ax.xaxis.label.set_fontsize(20)
ax.yaxis.label.set_fontsize(20)
apd_mad =
pd.DataFrame(data=list_mad_training,columns=["Mean
Absolute Distance (Train)"])
pd_mad["Mean Absolute Difference (Test)"] =
list_mad_testing;
pd_mad['Subdivisions'] = subdivs;
fig = plt.figure(figsize=(16,8))
ax = fig.add_subplot(111)
#pd_mad.groupby('Subdivisions').mean().plot(title='Overa
ll Rainfall in Each Month of Year', ax=ax,fontsize=20)
pd_mad.groupby('Subdivisions').mean().plot.bar(
width=0.5,title='Mean Absolute Difference for
Subdivisions',ax=ax, fontsize=20)
plt.xticks(rotation = 90)
plt.ylabel('Mean Abs. Difference')
ax.title.set_fontsize(30)
ax.xaxis.label.set_fontsize(20)
ax.yaxis.label.set_fontsize(20)

```

**8. Inference:**

It can be inferred that the error difference may just be about 75-76 % for the entire set if taken together. However these results may be justified as they are for predicting the rainfall for seven months given the data for five months for a period of over 100 years in 36 different regions. Additionally while plotting the the mean absolute differences between the test and train data in a bar chart we can see that the regions with higher annual rainfall are the regions where the residuals were the highest. From the bar plot of the mean annual rainfall of every region we can see that the six regions with the highest rainfall are Arunachal Pradesh, Coastal Karnataka , Kerala ,Konkan & Goa, Andaman and Nicobar Islands and the Sub-Himalayan Regions and Sikkim. In the bar chart of the residuals also it may be seen that the regions with highest mean absolute difference are the same regions with the highest rainfall. Other regions have a residual lesser than 75 which is the overall Mean Absolute Difference. Hence it can be safely assumed that the annual rainfall in higher rainfall regions may vary by higher values each year . In contrast regions with lower average rainfall have lesser variations in their annual rainfall and thus this linear regression model may work perfectly in those regions.



The high variations in the rainfall over the period in the given data can be seen in the (scatter plot:1) above. For example the region of Arunachal Pradesh experienced very high rainfall from 1925 to 1950 and later took a down slope. This could be a reason why the prediction of the rainfall in that region has the highest residue.

**References**

- Guhathakurta, P (2005) "Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network", Current Science, 90, 773-779
- Hastenrath, S (1988) "Prediction of Indian Monsoon Rainfall: Further Exploration", Journal of Climate, 1, 298-304.
- Abdouramane Gado Djibo 1, 2,\*, Harouna Karambiri 1, Ousmane Seidou 2, Ketvara Sittichok 2, Nathalie Philippon 3, Jean Emmanuel Paturel 4 and Hadiza Moussa Saley 5 (2015)
- "Linear and Non-Linear Approaches for Statistical Seasonal Rainfall Forecast in the Sirba Watershed Region (SAHEL)"Climate 2015, 3, 727-752; doi:10.3390/cli3030727