**Midterm Exam Report:**

**(1)Design:**

**Features:**

* Predicts **slump flow** of concrete based on the concrete ingredients.
* User needs to enter cement,slag,fly ash,water,SP,coarse and fine aggr values
* Slump flow is predicted based on input values
* It is a responsive application.Based on Twitter Bootstrap framework

**Integration algorithm:**

Written map-reduce program in python and refined the dataset

**Predictive model and algorithm:**

**Algorithm:**

Linear regression:It  is an approach for modeling the relationship between a scalar [dependent variable](http://en.wikipedia.org/wiki/Dependent_variable) y and one or more [explanatory variables](http://en.wikipedia.org/wiki/Explanatory_variable) denoted X.

**Model:**

Fitting Linear Models: **lm** is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance

**Datasets:**

UCI data repository-Concrete slump test data

<https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test>

**Web Design:**

It is a responsive application.Based on Twitter Bootstrap framework

**(2)Features implemented**

**Integrated algorithms:**

Written map reduce programs to refine dataset and convert .txt file to .csv file in python.

***Program to convert .txt to .csv****:*

import csv

import sys

with open(slump\_file, "rb") as infile, open(slump\_csv\_file, 'wb') as outfile:

in\_txt = csv.reader(infile, delimiter = ',')

out\_csv = csv.writer(outfile)

out\_csv.writerows(in\_txt)

**Predictive algorithms:**

**Linear regression** is an approach for modeling the relationship between a scalar [dependent variable](http://en.wikipedia.org/wiki/Dependent_variable) y and one or more [explanatory variables](http://en.wikipedia.org/wiki/Explanatory_variable) denoted X. The case of one explanatory variable is called [simple linear regression](http://en.wikipedia.org/wiki/Simple_linear_regression). For more than one explanatory variable, the process is called multiple linear regression. In linear regression, [data](http://en.wikipedia.org/wiki/Data) are modeled using [linear predictor functions](http://en.wikipedia.org/wiki/Linear_predictor_function), and unknown model [parameters](http://en.wikipedia.org/wiki/Parameters) are [estimated](http://en.wikipedia.org/wiki/Estimation_theory) from the data. Such models are called [linear models](http://en.wikipedia.org/wiki/Linear_model). Most commonly, linear regression refers to a model in which the [conditional mean](http://en.wikipedia.org/wiki/Conditional_expectation) of y given the value of X is an [affine function](http://en.wikipedia.org/wiki/Affine_transformation) of X. Less commonly, linear regression could refer to a model in which the [median](http://en.wikipedia.org/wiki/Median), or some other [quantile](http://en.wikipedia.org/wiki/Quantile) of the conditional distribution of y given X is expressed as a linear function of X. Like all forms of [regression analysis](http://en.wikipedia.org/wiki/Regression_analysis), linear regression focuses on the [conditional probability distribution](http://en.wikipedia.org/wiki/Conditional_probability_distribution) of y given X, rather than on the [joint probability distribution](http://en.wikipedia.org/wiki/Joint_probability_distribution) of y and X, which is the domain of [multivariate analysis](http://en.wikipedia.org/wiki/Multivariate_analysis).

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

* If the goal is prediction, or forecasting, or reduction, linear regression can be used to fit a predictive model to an observed data set of y and X values. After developing such a model, if an additional value of X is then given without its accompanying value of y, the fitted model can be used to make a prediction of the value of y.
* Given a variable y and a number of variables X1, ..., Xp that may be related to y, linear regression analysis can be applied to quantify the strength of the relationship between y and the Xj, to assess which Xj may have no relationship with y at all, and to identify which subsets of the Xj contain redundant information about y.

**Predictive model:**

## Fitting Linear Models

### *Description*

### lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although [aov](http://127.0.0.1:29385/help/library/stats/help/aov) may provide a more convenient interface for these).

### *Usage*

lm(formula, data, subset, weights, na.action,

method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,

singular.ok = TRUE, contrasts = NULL, offset, ...)

### *Arguments*

formula: an object of class "[formula](http://127.0.0.1:27665/help/library/stats/help/formula)" (or one that can be coerced to that class): a symbolic description of the model to be fitted

data: an optional data frame, list or environment (or object coercible by [as.data.frame](http://127.0.0.1:27665/help/library/stats/help/as.data.frame) to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.

subsets: an optional vector specifying a subset of observations to be used in the fitting process.

weights: an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used with weights weights (that is, minimizing sum(w\*e^2)); otherwise ordinary least squares is used.

na.action: a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of [options](http://127.0.0.1:27665/help/library/stats/help/options), and is [na.fail](http://127.0.0.1:27665/help/library/stats/help/na.fail) if that is unset. The ‘factory-fresh’ default is [na.omit](http://127.0.0.1:27665/help/library/stats/help/na.omit). Another possible value is NULL, no action. Value [na.exclude](http://127.0.0.1:27665/help/library/stats/help/na.exclude)can be useful.

method: the method to be used; for fitting, currently only method = "qr" is supported; method = "model.frame" returns the model frame

model,x,y,qr: ogicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.

singular.ok: logical. If FALSE (the default in S but not in R) a singular fit is an error.

contrasts: an optional list. See the contrasts.arg of [model.matrix.default](http://127.0.0.1:27665/help/library/stats/help/model.matrix.default).

offset: this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more [offset](http://127.0.0.1:27665/help/library/stats/help/offset) terms can be included in the formula instead or as well, and if more than one are specified their sum is used.

***Linear model for this application:***

*SLUMP.cm. = c0 + c1(Cement)+c2(Slag)+c3(Fly.ash)+c4(Water)+c5(SP)+c6(Coarse.Aggr.)+c7(Fine.Aggr.)*

Run multivariate Linear regression algorithm using R on dataset

***Script in R:***

/\*Store csv file data in slump\*/

slump <- read.csv("C:\\Users\\Aditya Deshpande\\Downloads\\slump.csv")

/\*Displays number of columns in dataset\*/

str(slump)

/\*Displays summary of data like median,mean,min and high values\*/

summary(slump)

?lm

/\*multivariate linear regression\*/

LevLevReg <- lm(SLUMP.cm.~Cement + Slag +Fly.ash+Water+SP+Coarse.Aggr.+Fine.Aggr.,slump)

/\*Displays coefficients of linear regression\*/

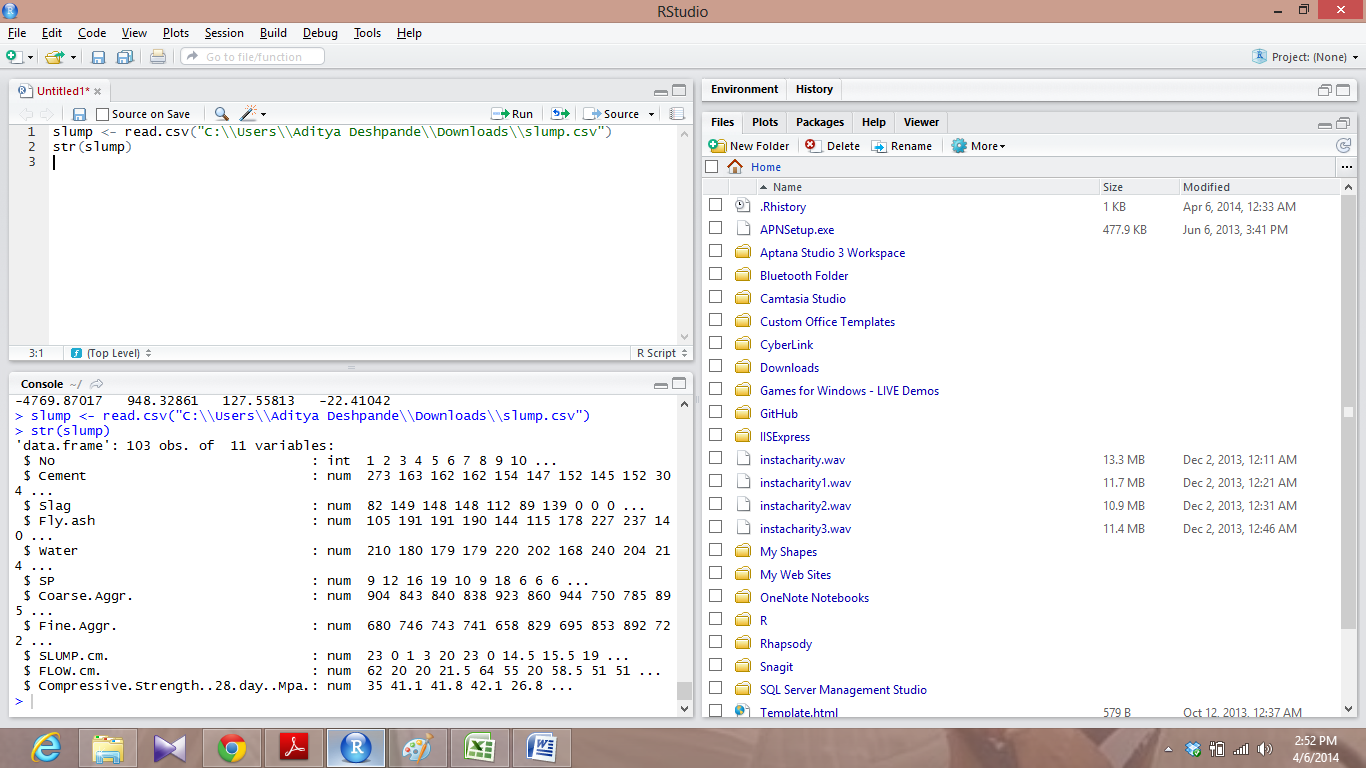
coefficients(LevLevReg)

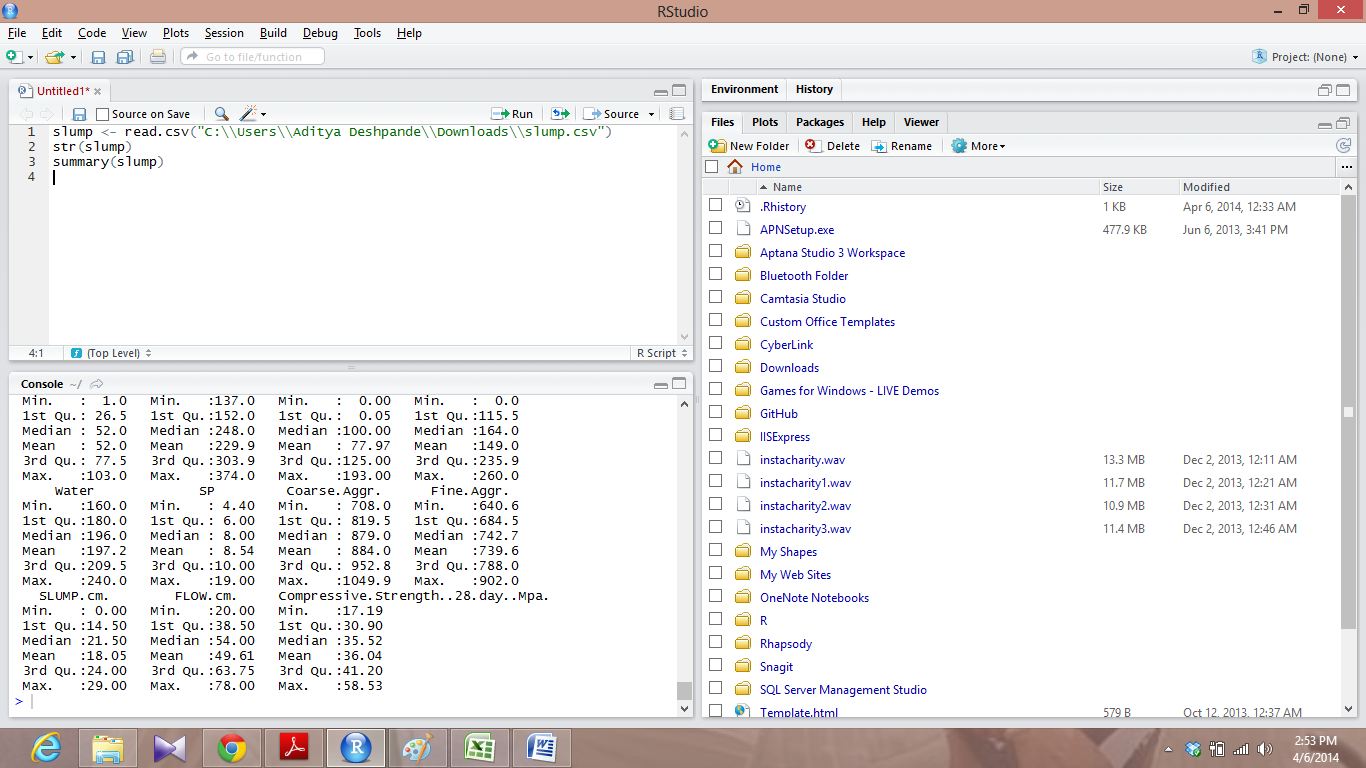
/\*Calculates original values based on coefficients and intercept\*/

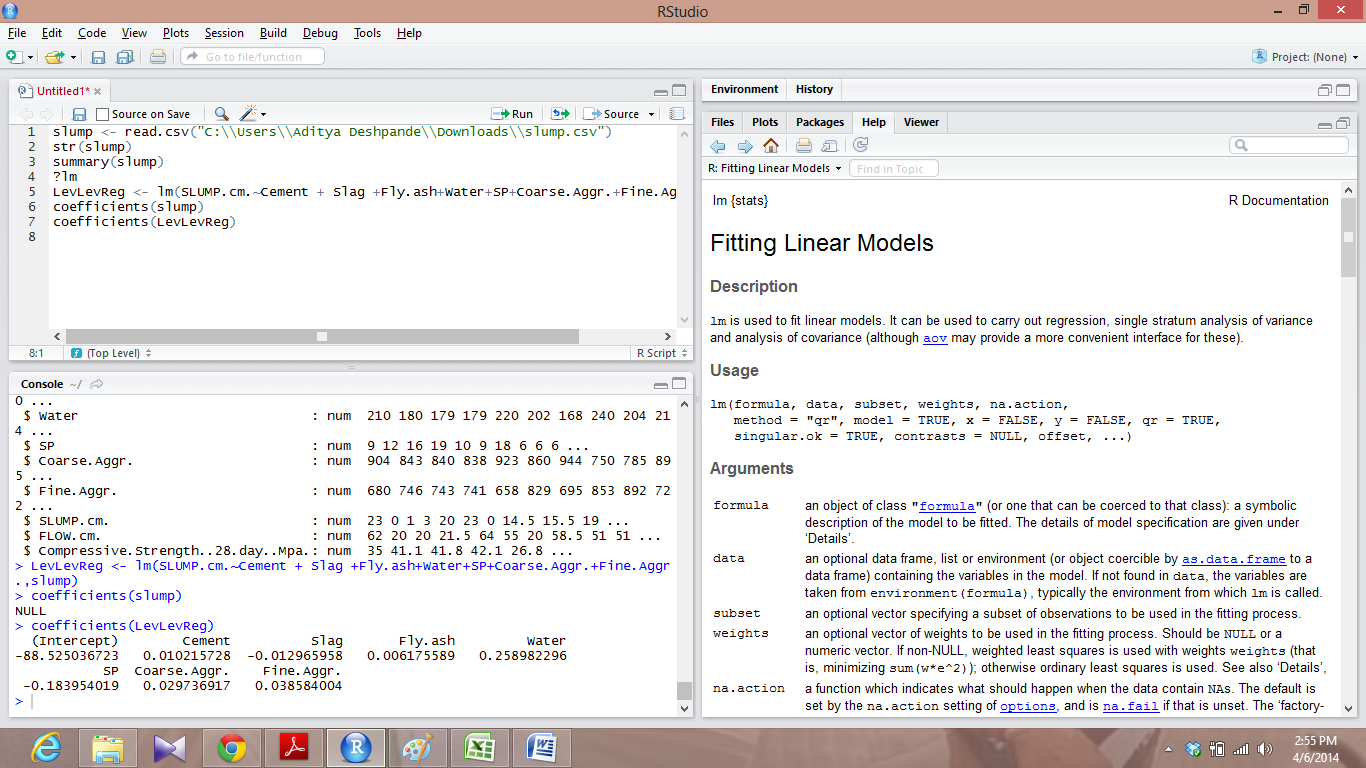
fitted.values(LevLevReg)

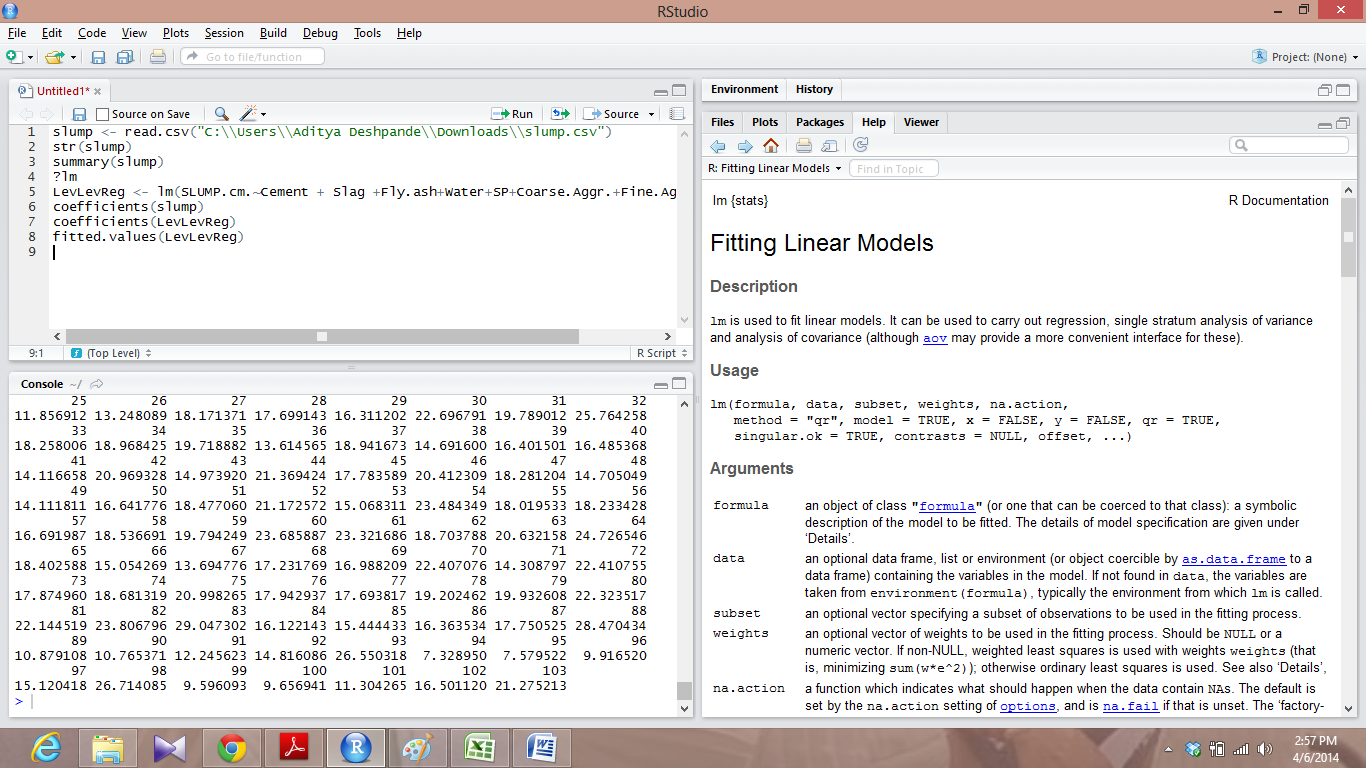
/\*Display Plots \*/

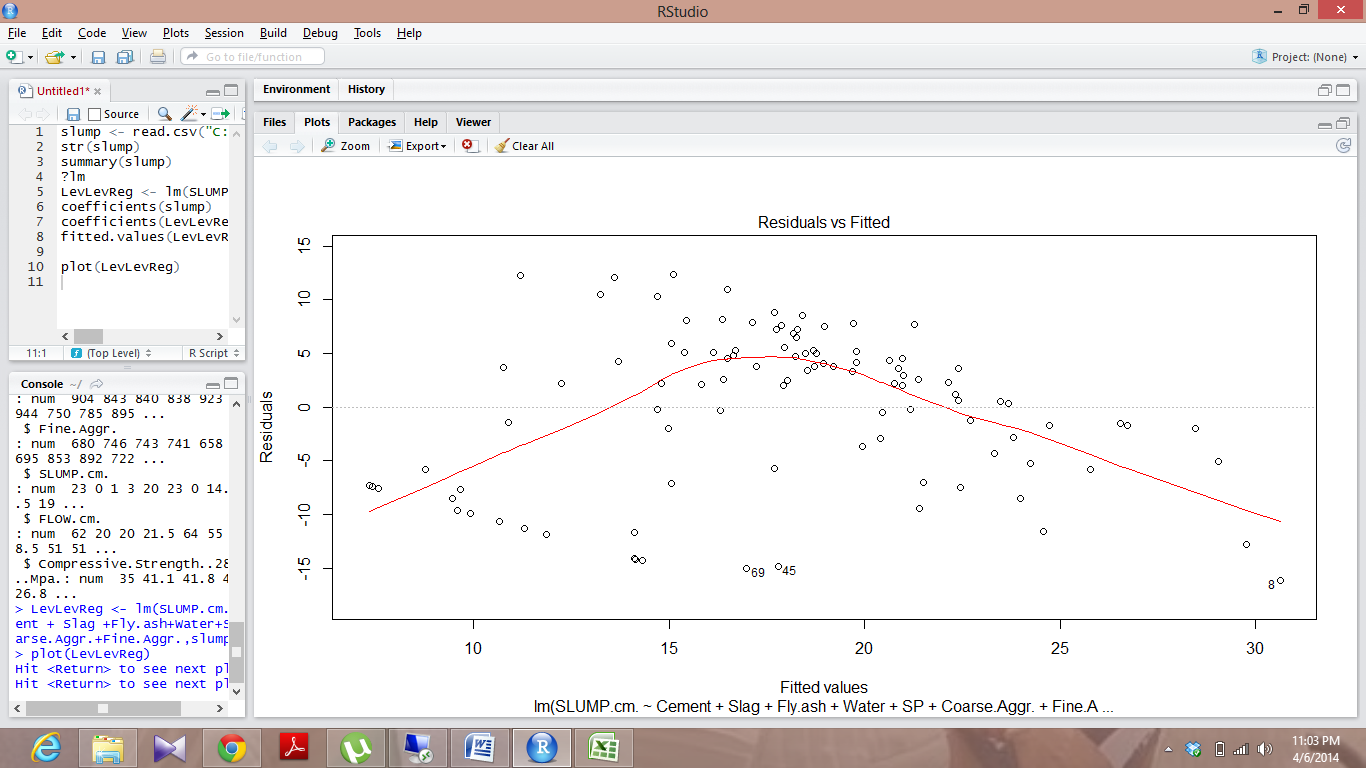
plot(LevLevReg)

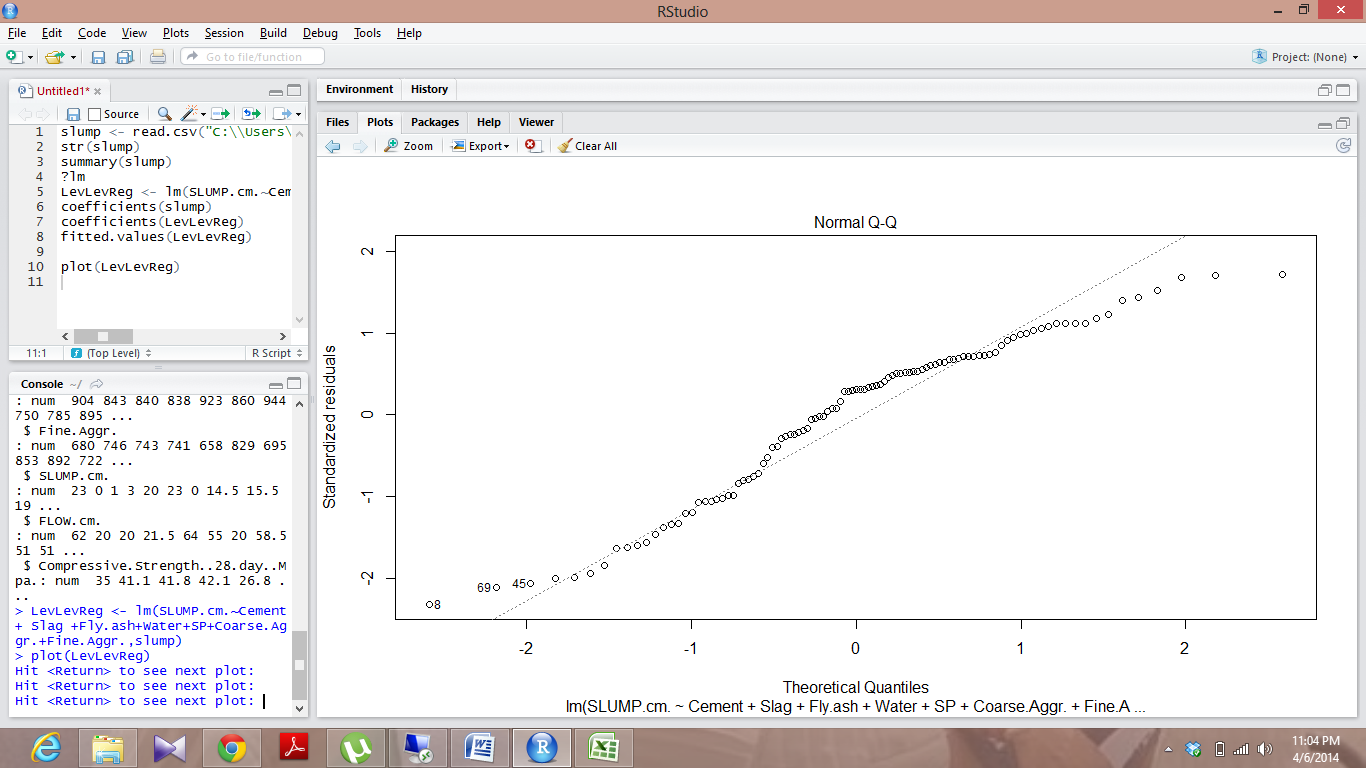


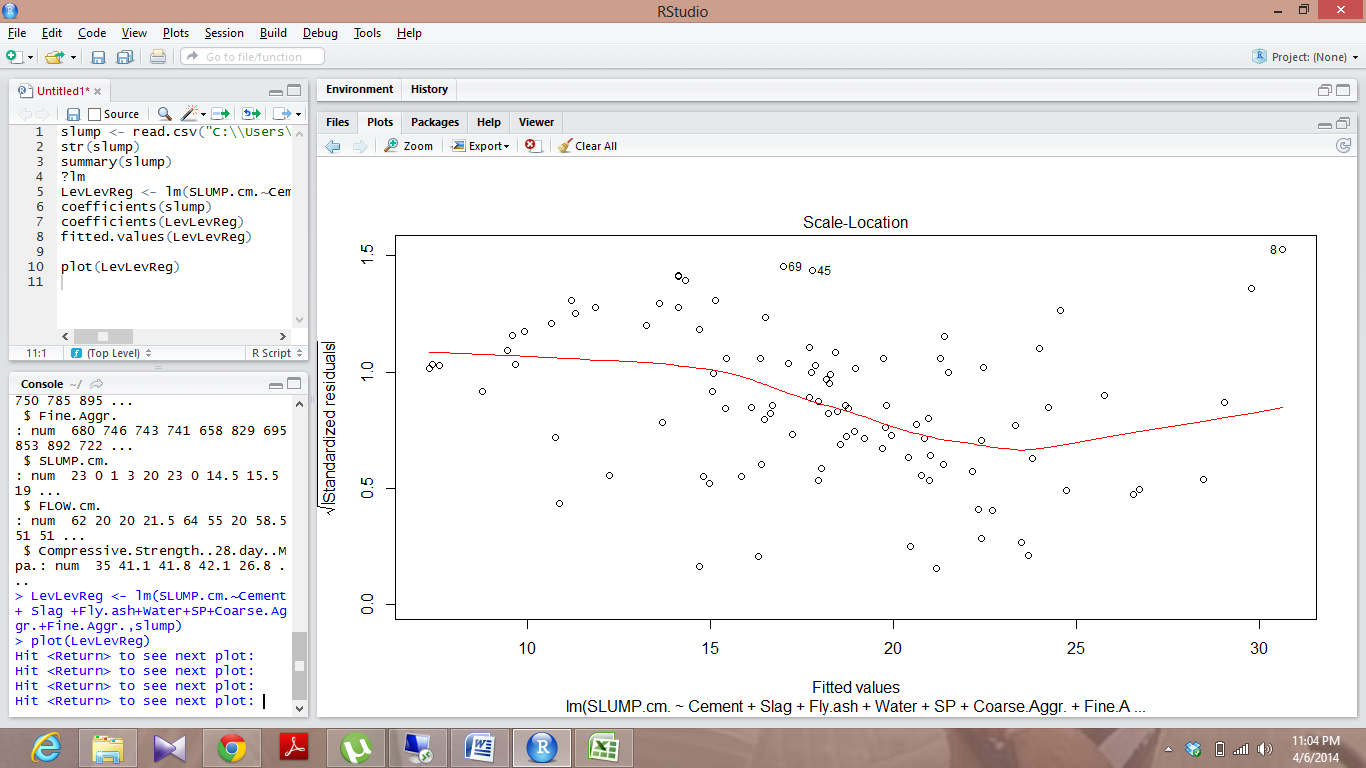


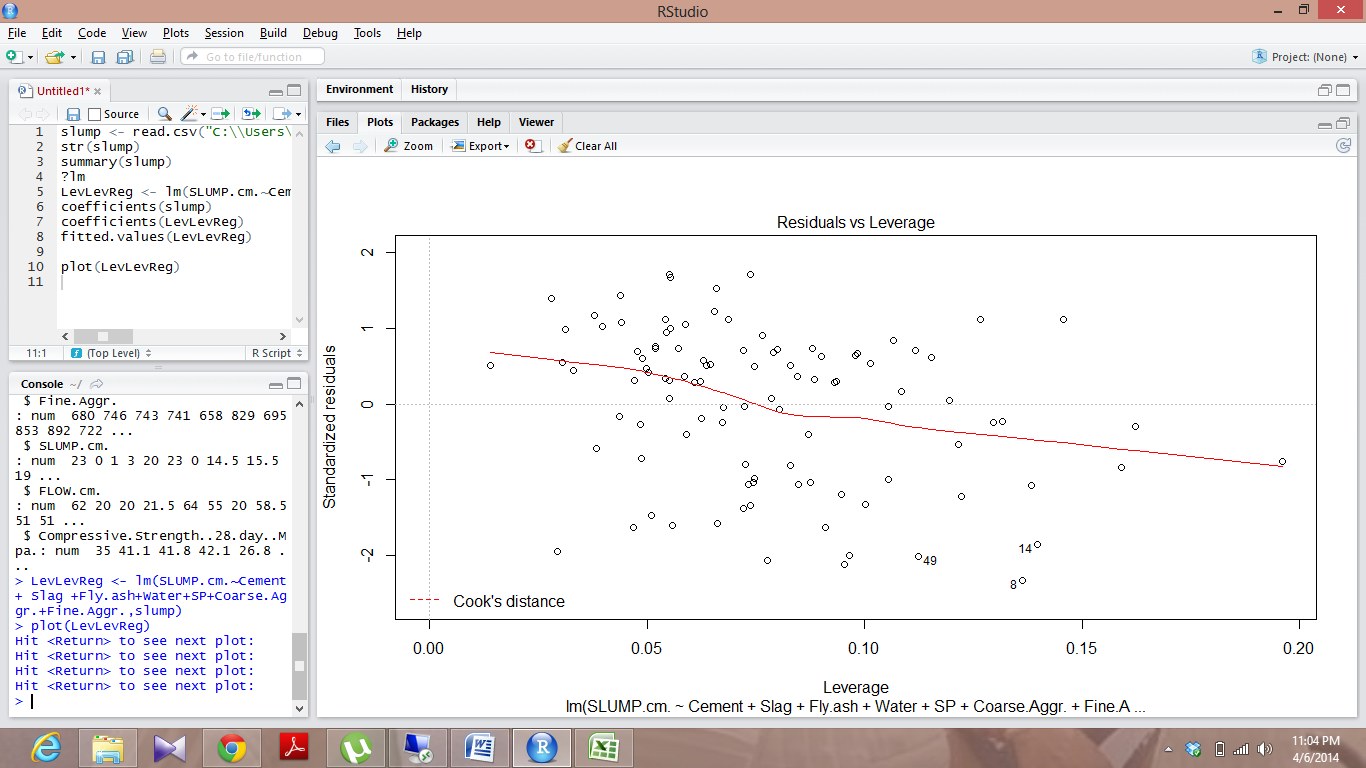












**Solr indexing:**

*Service URL:*

<http://134.193.136.127:8983/solr/collection1_shard1_replica1/select?q=id%3AAditya&wt=json&indent=true>

Service would return coefficients and intercepts values for this linear model i.e; c0,c1,c2,c3,c4,c5,c6 and c7.

*JSON format*

{

"responseHeader":{

"status":0,

"QTime":1,

"params":{

"indent":"true",

"q":"id:Aditya",

"wt":"json"}},

"response":{"numFound":1,"start":0,"docs":[

{

"id":"Aditya",

"title":["-88.525036723",

"0.010215728",

"-0.012965958",

"0.006175589",

"0.258982296",

"-0.183954019",

"0.029736917",

"0.038584004"],

"\_version\_":1464675360509526016}]

}}

*Python Program to convert .csv to json*

import csv

import json

csvf = open('slump.csv', 'r')

jsonf = open('slump.json', 'w')

field\_Names = ("id","title")

reader = csv.DictReader( csvf, field\_Names)

for row in reader:

json.dump(row, jsonf)

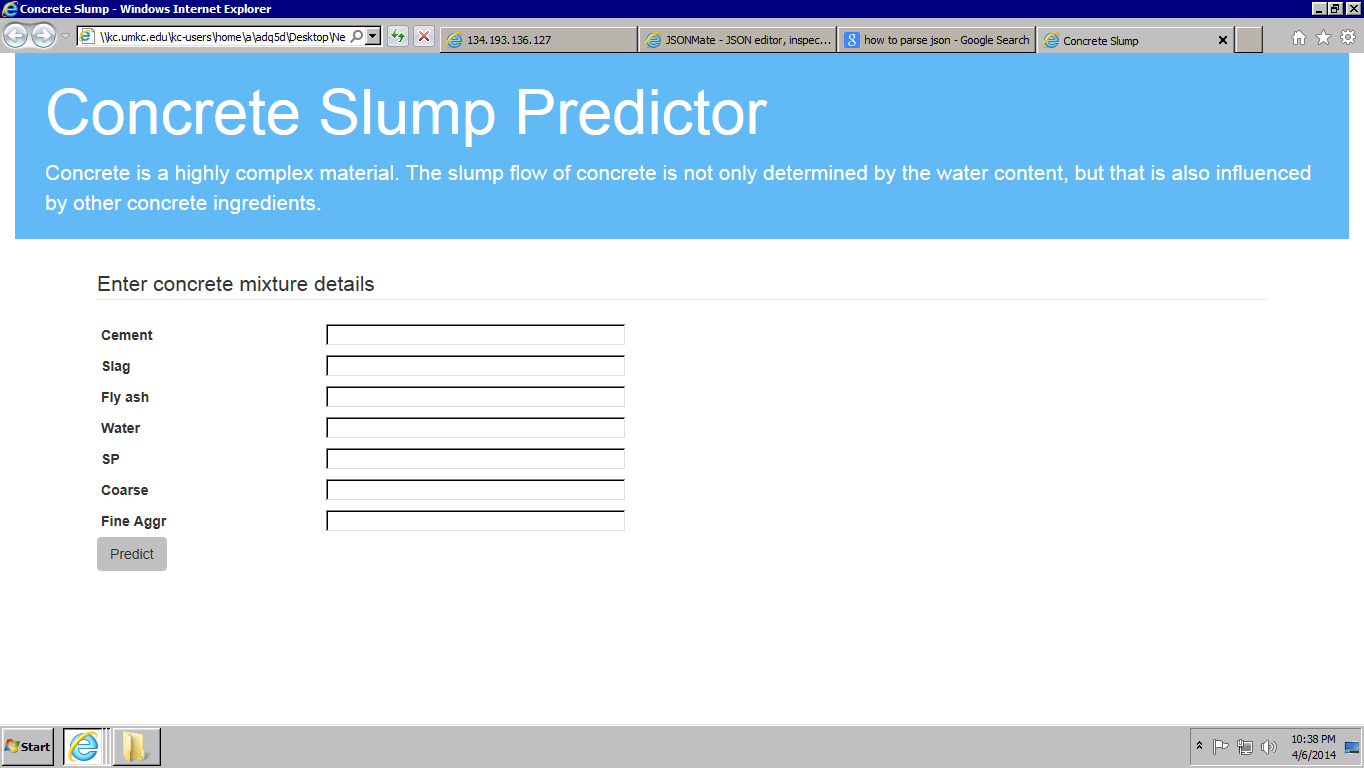
jsonfile.write('\n')

***Mobile User Interface***

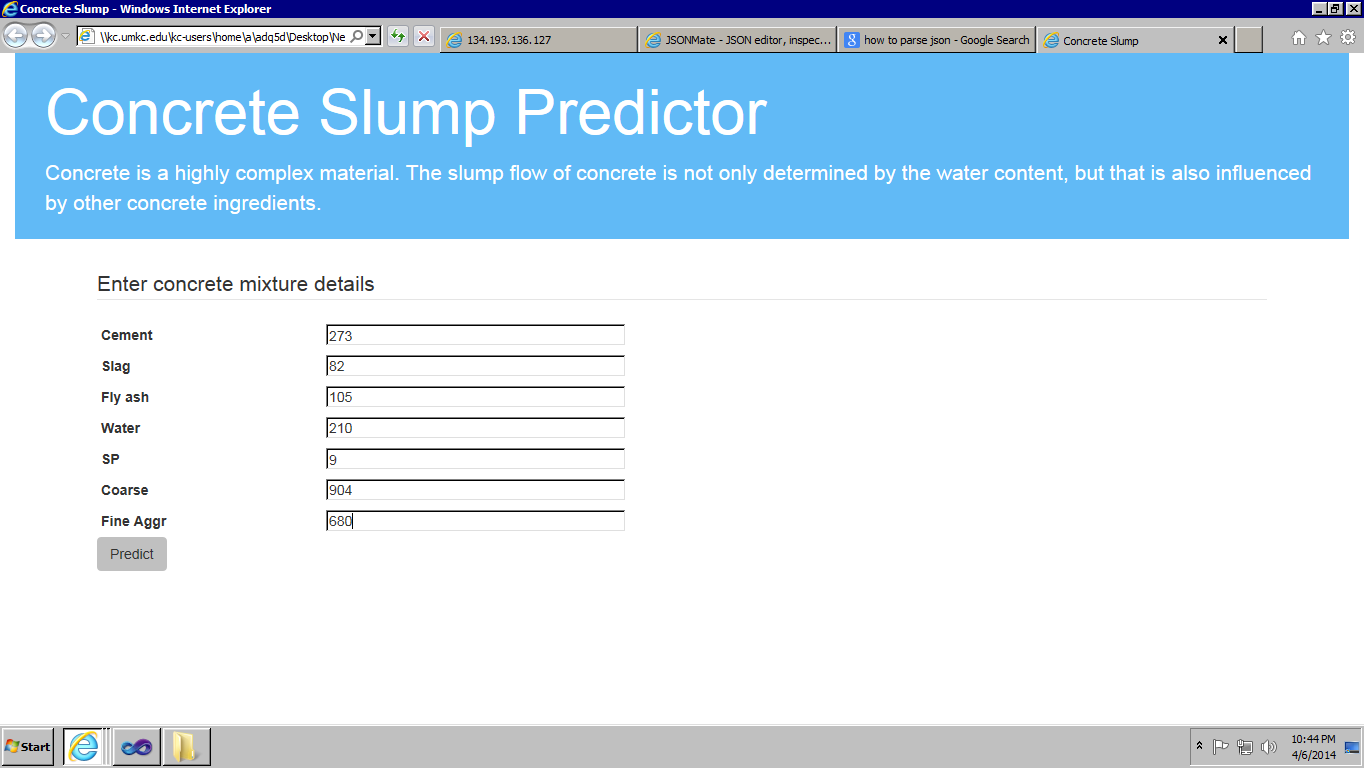
Used Twitter Bootstrap to design the User Interface

**Outputs:**

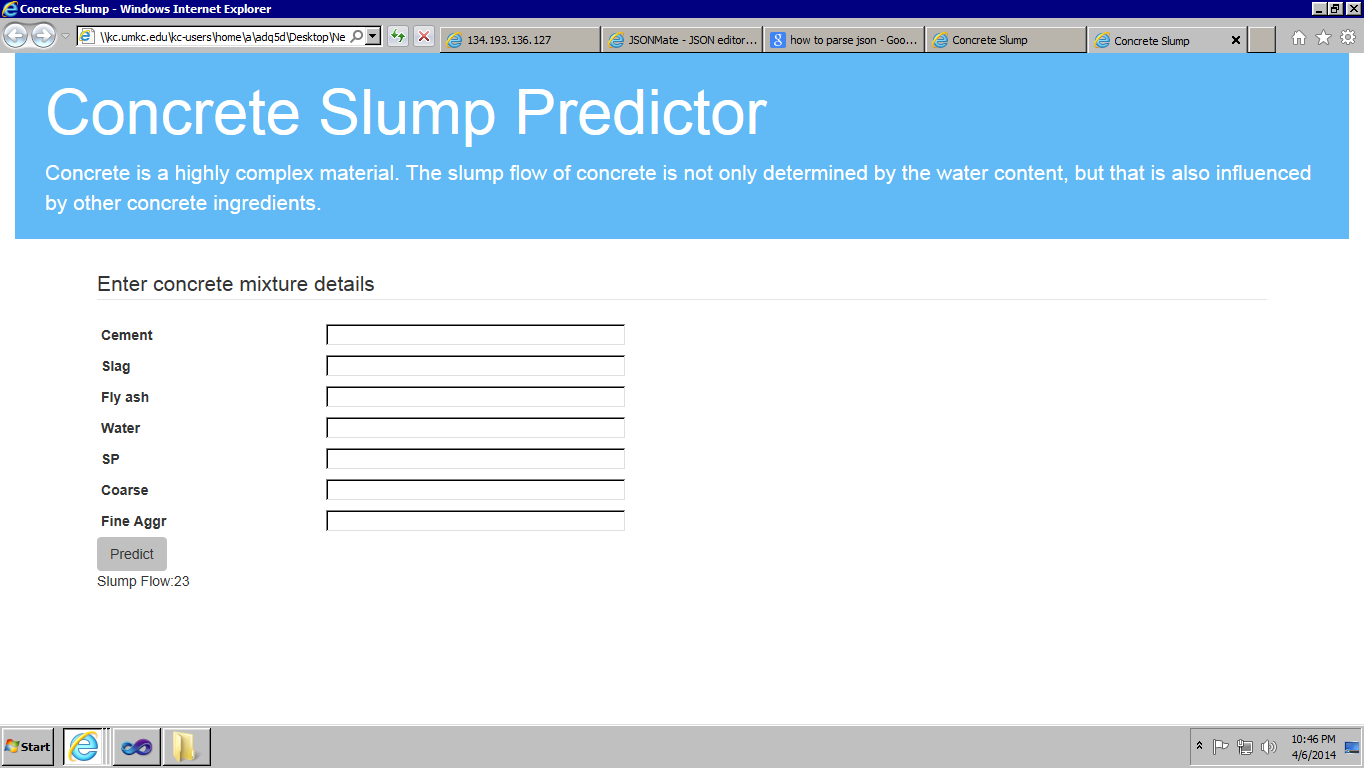
*Home screen*



Enter values to predict slump flow



Predicts the values based on values given



**Website URL:**

[\\kc.umkc.edu\kc-users\home\a\adq5d\Desktop\New folder (2)\Site\index.html](file:///\\kc.umkc.edu\kc-users\home\a\adq5d\Desktop\New%20folder%20(2)\Site\index.html)

**Web service URL:**

<http://134.193.136.127:8983/solr/collection1_shard1_replica1/select?q=id%3AAditya&wt=json&indent=true>

**GitHub URL:**

**Limitations:**

It only predicts numeric values

Works efficiently with .csv files

**References:**

<http://stackoverflow.com/>

<http://archive.ics.uci.edu/ml/>

<http://www.r-project.org/>