INDIAN INSTITUTE OF TECHNOLOGY

BANARAS HINDU UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

A Project report on

Exploratory Data Analysis

Submitted at the conclusion of

Summer Internship

(9 May to 23 June 2012)

ACKNOWLEDGEMENT

It has been indeed a great privilege for me to have Prof. K. K. Shukla, Department of Computer Science and Engineering, Indian Institute of Technology, Banaras Hindu University, as my training supervisor. His awe-inspiring personality, superb guidance and constant encouragement were the motive force behind this project work.

I am also thankful to all the technical and non – teaching staff of the department for their constant assistance and co – operation.

Alankrita

2009UIT107

IIIrd Year,Information Technology

MNIT,Jaipur

DEPARTMENT OF COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
BANARAS HINDU UNIVERSITY
VARANASI-221005, INDIA

Dr. K.K.Shukla
Professor
Computer Science and Engineering

Tel # +91-542-2307056
Fax # +91-542-2368428
Email-kkshukla@bhu.ac.in

Ref. No. IT/CSE/2011-12/ Date-05.07.2012

CERTIFICATE

This is to certify that Alankrita, Roll No:2009UIT107 student of Information Technology, Malaviya National Institute Of Technology,Jaipur has successfully completed her summer training project titled

"EXPLORATORY DATA ANALYSIS"

Under the supervision of Prof.K.K Shukla,Computer Engineering Department,IIT BHU. The report submitted by her embodies the literature from various sources and from the material provided by me during the period.

Prof. K.K. Shukla
Dated: 05.07.2012
Supervisor
Department of Computer Engineering
Indian Institute of Technology, BHU

Table of Contents

1.INTRODUCTION:

In statistics, exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics in easy-to-understand form, often with visual graphs, without using a statistical model or having formulated a hypothesis. Exploratory data analysis was promoted by John Tukey to encourage statisticians visually to examine their data sets, to formulate hypotheses that could be tested on new data-sets.It is mostly a philosophy of data_analysis where the researcher examines the data without any pre-conceived_ideas in order to discover what the data can tell him about the phenomena_being studied. Exploratory Data Analysis employs a variety of techniques (mostly graphical) to

- Maximize insight into a dataset
- Uncover the underlying structure
- Extract important variables
- Test underlying assumptions
- Develop parsimonious models

Some of the typical graphical techniques used in EDA are:

- Principal component analysis
- Histogram
- Scatter plot
- Box plot
- Multi-vari chart
- Multi dimensional scaling

The objective of the report is to describe the some of the graphical techniques used in EDA with suitable examples and experiments and present a detailed analysis for the same.

2. DESCRIPTION OF ATTRIBUTES AND DATASETS USED

Attribute name	Type	Symbol	two	three	Four	Five	Six	Seven
ESSENTIAL_COMPLEXITY	numeric		√	√		√	√	√
ESSENTIAL_DENSITY	numeric		√	√		√	√	√
LOC_EXECUTABLE	numeric		√					
PARAMETER_COUNT	Numeric		√		√			
GLOBAL_DATA_COMPLEXITY	Numeric		√					
GLOBAL_DATA_DENSITY	Numeric		√			√		
HALSTEAD_CONTENT	Numeric		√	√		√	√	√
HALSTEAD_DIFFICULTY	Numeric	d	√	√	√	√	√	√
HALSTEAD_EFFORT	Numeric	e	√	√	√	√	√	√
HALSTEAD_ERROR_EST	Numeric		√	√	√	√	√	√
HALSTEAD_LENGTH	numeric	l	√	√	√	√	√	√
HALSTEAD_LEVEL	numeric		√	√	√	√	√	√
HALSTEAD_PROG_TIME	numeric		√	√	√	√	√	√
HALSTEAD_VOLUME	numeric	v	√		√	√	√	√
MAINTENANCE_SEVERITY	numeric		√	√	√	√	√	√
MULTIPLE_CONDITION_COUNT	numeric		√	√	√	√	√	√
NODE_COUNT	numeric		√	√			√	√
NORMALIZED_CYLOMATIC_COMPLEXITY	numeric		√	√	√	√	√	√
LOC_TOTAL	numeric	loc	√	√	√	√	√	√
NUM_OPERANDS	numeric	n	√	√	√	√		√
NUM_OPERATORS	numeric		√	√	√	√	√	√
NUM_UNIQUE_OPERANDS	numeric		√	√	√	√	√	√
C	False/true		√	√	√	√		√
LOC_BLANK	numeric		√	√	√	√	√	√
BRANCH COUNT	numeric		√	√	√	√		√
LOC_COMMENT	numeric			√	√	√	√	√
DESIGN DENSITY	numeric			√			√	√
EDGE COUNT	numeric	V(g)		√				√
cyclomatic complexity	numeric			√	√	√	√	`√
DECISION COMPLEXITY	numeric		√	√		√	√	√

*two,three,four,five,six and seven are the names given to the different datasets.

2.1 KURTOSIS ANALYSIS ON THE DATASETS

DATASET TWO

Following is the result of kurtosis function on the data set. Result shows that none of the attributes are normally distributed since the kurtosis of the given dataset is less than three.

```
Columns 1 through 10

 0.0130    0.3750    0.0129    0.0603    0.0180    0.4490    0.3579    0.0075    0.4984    0.0022

  Columns 11 through 20

  0.0731    0.0018    0.3452    0.0721    0.0039    0.1592    0.0042    0.2706    0.0666    1.0390

  Columns 21 through 30

 0.5629    0.4183    0.0345    1.0390    0.5628    0.0015    0.3996    0.4390    0.3351    0.0240

  Columns 31 through 37

 0.3452    0.4851    0.2090    0.0073    0.0798    0.0041    0.1468
```

DATASET FOUR:

The result of kurtosis function shows that the distribution is more outlier prone to the normal distribution .Since the value of kurtosis for all the attributes are greater than three.

```
Columns 1 through 10

  101.6676    67.5013   221.7819   152.0962    60.9105   112.1843    32.9723    62.2601    69.8413
411.6345

 Columns 11 through 20

  110.7598   411.6346  109.0448    53.6046    93.3026   104.1594    16.6629   119.0996    71.3971
49.1999

 Column 21

 57.1487
```

DATASET FIVE

Following is the result of the kurtosis function on the data set. The result shows that the distribution is less outlier prone to the normal distribution**.**

```
 Columns 1 through 10

 0.0130    0.3750    0.0129    0.0603    0.0180    0.4490    0.3579    0.0075    0.4984    0.0022

 Columns 11 through 20

 0.0731    0.0018    0.3452    0.0721    0.0039    0.1592    0.0042    0.2706    0.0666    1.0390

 Columns 21 through 30

0.5629    0.4183    0.0345    1.0390    0.5628    0.0015    0.3996    0.4390    0.3351    0.0240

Columns 31 through 37

0.3452    0.4851    0.2090    0.0073    0.0798    0.0041    0.1468
```

DATASET SIX

Following result of kurtosis function show that none of the attributes are normally distributed. They are less outlier prone to the normal distribution.

```
Columns 1 through 10

  0.0130    0.3750    0.0129    0.0603    0.0180    0.4490    0.3579    0.0075    0.4984
0.0022

 Columns 11 through 20

 0.0731    0.0018    0.3452    0.0721    0.0039    0.1592    0.0042    0.2706    0.0666
1.0390

 Columns 21 through 30

  0.5629    0.4183    0.0345    1.0390    0.5628    0.0015    0.3996    0.4390    0.3351
0.0240

 Columns 31 through 37
 0.3452    0.4851    0.2090    0.0073    0.0798    0.0041    0.1468
```

DATASET: Seven

```
Columns 1 through 10

18.2835    55.5878    13.1658    50.8812    17.3112   184.4417    49.1038   267.8278
246.0472    1.7979

 Columns 11 through 20

 87.6100     1.8820    44.9179    91.9243    12.2314    14.9058     5.9477 771.9855
10.3509  184.8189

  Columns 21 through 30

 194.1911  102.3522    9.2602  184.8189  194.1274    1.4901  136.5247  168.1199
46.1952   28.2382

Columns 31 through 37

 145.9701   82.2210  470.5350    3.7662   68.3695    3.4269   15.1223
```

3.DIMENTION REDUCTION:

Dimensionality reduction is the process of finding a suitable lower dimensional space in which to represent the original data. Our goal is that the alternative representation of the data will help us:

- Visualize the data
- Explore high-dimensional data with the goal of discovering structure or patterns that lead to the formation of statistical hypotheses.
- using scatter plots when dimensionality is reduced to 2-D or 3-D.

- Analyze the data using statistical methods, such as clustering smoothing, probability density estimation, or classification.

3.1PRINCIPAL COMPONENT ANALYSIS

The main purpose of **principal component analysis** (PCA) is to reduce the dimensionality from *p* to *d*, where *d* < *p*, while at the same time accounting for as much of the variation in the original data set as possible. With PCA, we transform the data to a new set of coordinates or variables that are a linear combination of the original variables.

Before getting to a description of PCA, It is necessary to describe the basic mathematical concepts that will be used in PCA. It covers standard deviation, covariance, eigenvectors and eigenvalues.

3.1.1STANDARD DEVIATION

To understand standard deviation, we need a data set. Consider an example set

X=[1 2 4 6 12 15 25 45 68 67 65 98]

We can calculate the mean of the data but the mean of the data only does not convey the enough information about the data.For example consider the two datasets that have same mean

[0 8 12 20] and [8 9 11 12]

So what is different about these two sets? It is the spread of the data that is different. The Standard Deviation (SD) of a data set is a measure of how spread out the data is. How do we calculate it? The English definition of the SD is: “The average distance from the mean of the data set to a point”. The way to calculate it is to compute the squares of the distance from each data point to the mean of the set, add them all up,

divide byn-1, and take the positive square root .

3.1.2 COVARIANCE

Many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions. For example, we might have as our data set both the height of all the students in a class, and the mark they received for that paper. We could then perform statistical analysis to see if the height of a student has any effect on their mark. Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set independently of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean with respect to each other. Covariance is such a measure. Covariance is always measured between 2 dimensions. If you calculate the covariance between one dimension and itself, you get the variance. So, if you had a 3-dimensional data set (x,y,z), then you could measure the covariance between the x and y dimensions, the y and z dimensions, and the z and x dimensions. Measuring the covariance between x and x, or y and y would give you the variance.

3.1.3 EIGEN VALUES AND EIGEN VECTORS

The eigenvectors of a square matrix are the non-zero vectors that, after being multiplied by the matrix, remain parallel to the original vector. For each eigenvector, the corresponding eigenvalue is the factor by which the eigenvector is scaled when multiplied by the matrix.

3.2 SCREE PLOT

A graphical way of determining the number of PCs to retain is called the **scree Plot.** The original name and idea is from Cattell [1966], and it is a plot of *lk* (the eigenvalue) versus *k* (the index of the eigenvalue). A Scree Plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each PC. The PCs are ordered, and by definition are therefore assigned a number label, by decreasing order of contribution to total variance. Such a plot when read left-to-right across the abscissa can often show a clear separation in fraction of total variance where the 'most important' components cease and the 'least important' components begin. The point of separation is often called the 'elbow'.

In some cases, we might plot the log of the eigenvalues when the first eigenvalues are very large. This type of plot is called a ***log-eigenvalue*** or LEV plot. To use the scree plot, one looks for the ‘elbow’ in the curve or the place where the curve levels off and becomes almost flat. Another way to look at this is by the slopes of the lines connecting the points. When the slopes start to level off and become less steep, that is the number of PCs one should keep.

Following is a MATLAB code showing PCA using covariance matrix. The **eig** function is used to calculate the eigenvalues and eigenvectors. MATLAB returns the eigenvalues in a diagonal matrix, and they are in ascending order, so they must be flipped to get the scree plot.

```
[n,p] = size(data);
% Center the data.
datac = data - repmat(sum(data)/n,n,1);
% Find the covariance matrix.
```

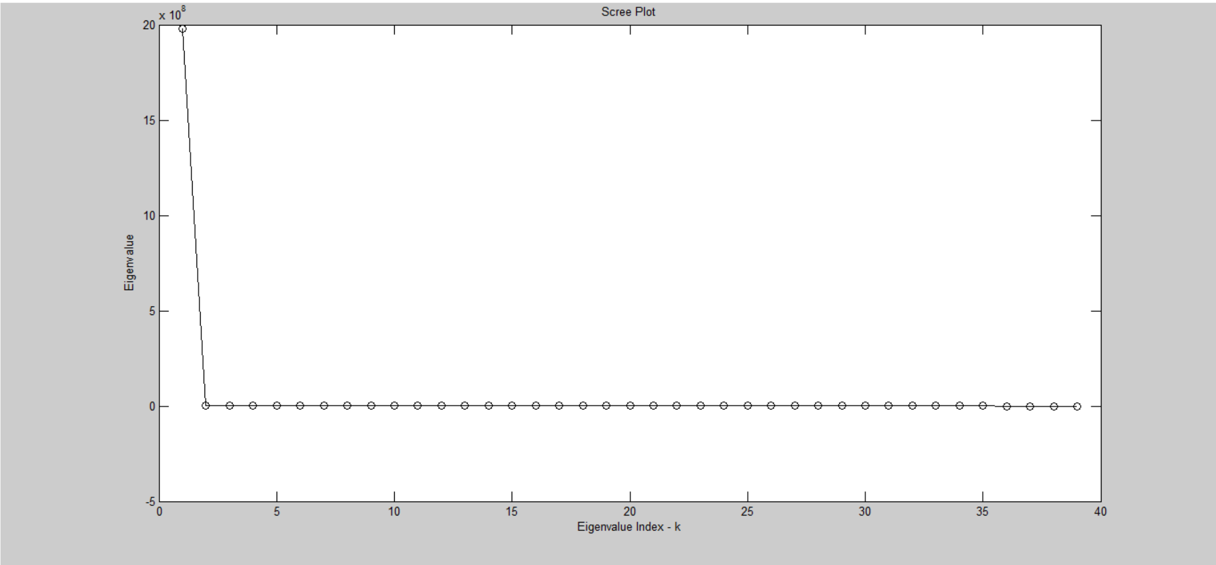
```
covm = cov(datac);

[eigvec,eigval] = eig(covm);
eigval = diag(eigval); % Extract the diagonal elements
% Order in descending order
eigval = flipud(eigval);
eigvec = eigvec(:,p:-1:1);

figure, plot(1:length(eigval),eigval,'ko-')
title('Scree Plot')
xlabel('Eigenvalue Index - k')
ylabel('Eigenvalue')
```

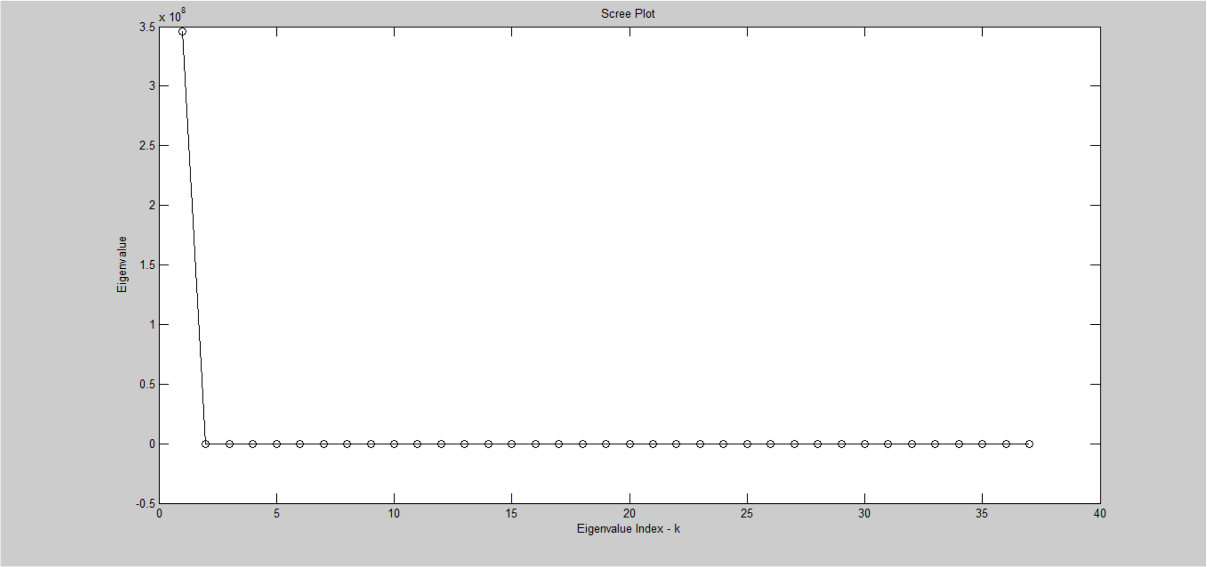
EXPERIMENT: Above MATLAB code was run using the following data sets

DATASET:TWO



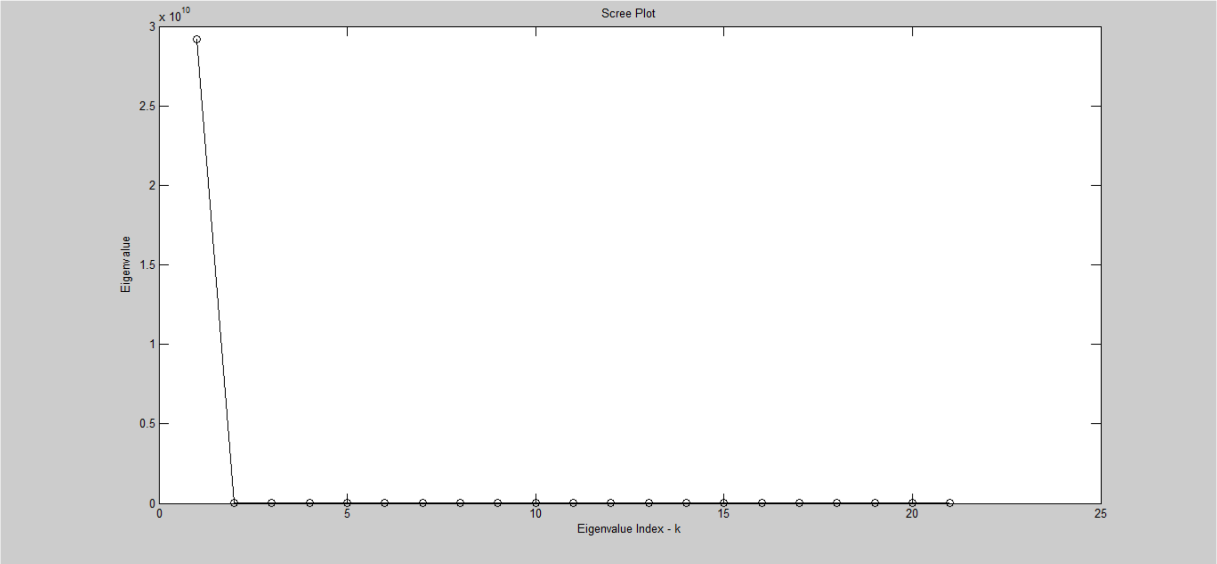
RESULT: Scree plot shows that for the dataset two which has 41 attributes, if we keep the number of pc's two, it can cover the whole variance of the data

DATASET: THREE

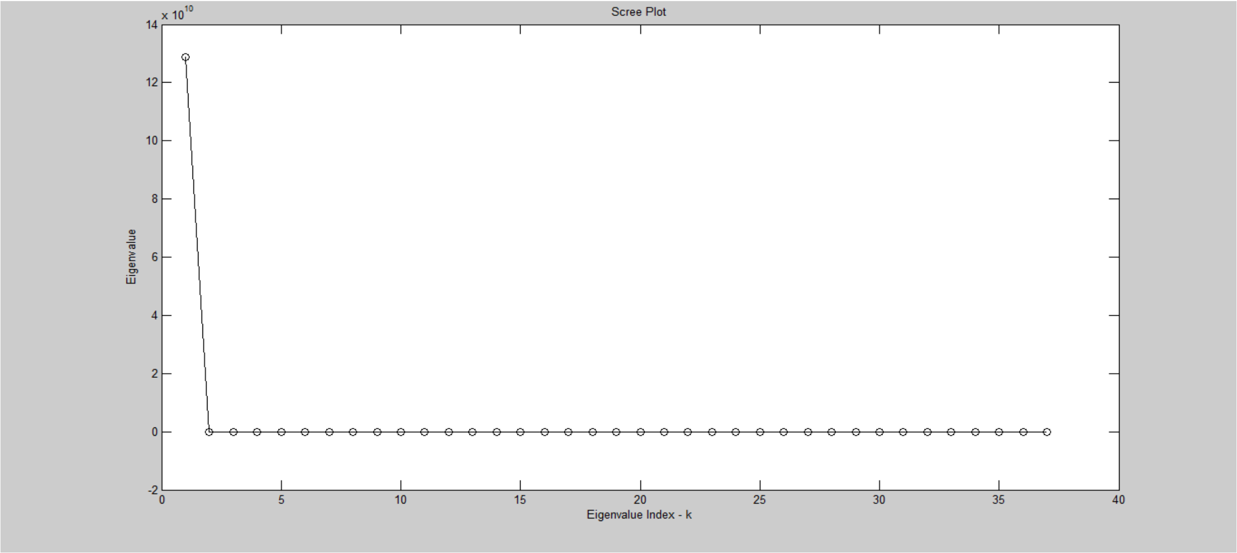


Result: The elbow is near 2. So the subsequent factors can be ignored and dimensionality of the data can be reduced to two for the dataset 'Three' which has 38 attributes.

DATASET FOUR:

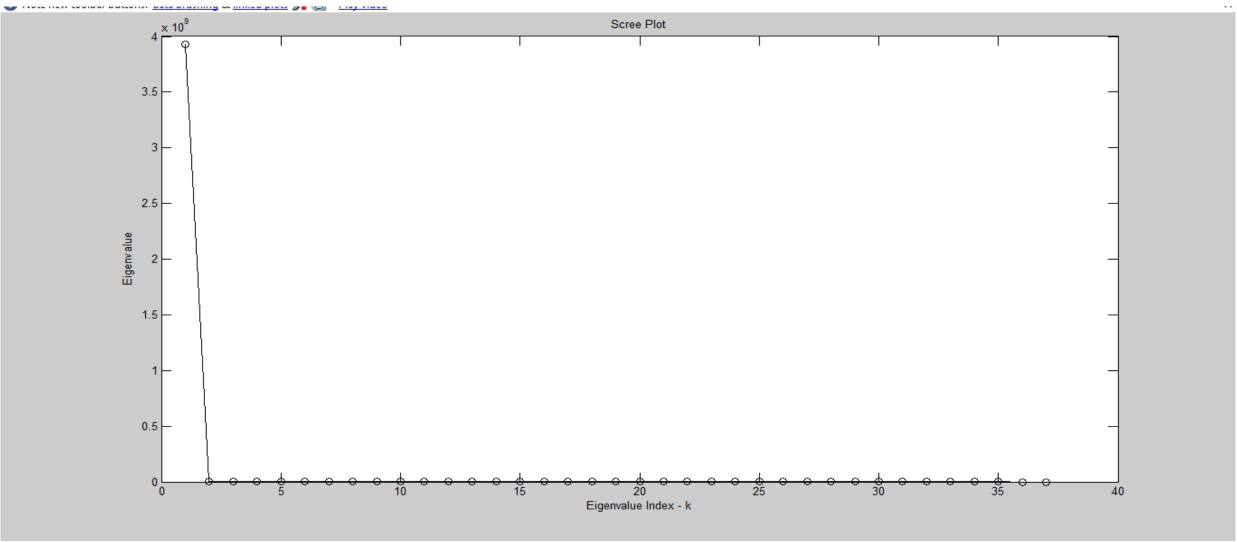


DATASET FIVE:



Result: The elbow is at 2.Thus for the dataset five which has 38 attributes for that the dimensionality can be reduced till two so that the whole variance of the dataset can be covered.

Dataset: Seven



Result: The dataset seven which has 38 attributes can be reduced two approximately two dimensions. As it can be seen from the scree plot that elbow of the graph is at two. Therefore the subsequent factors can be ignored.

4. DISTRIBUTION SHAPES

The ability to visualize the distribution shape in exploratory data analysis is important for several reasons. First, we can use it to summarize a data set to better understand general characteristics such as shape, spread, or location. In turn, this information can be used to suggest transformations or probabilistic models for the data. Second, we can use these methods to check model assumptions, such as symmetry, normality, etc. We present several techniques for visualizing univariate and bivariate distributions. These include 1-D and 2-D histograms, boxplots, quantilebased plots, and bagplots.

4.1 HISTOGRAMS

A **histogram** is a way to graphically summarize or describe a data set by visually conveying its distribution using vertical bars. They are easy to create and are computationally feasible, so they can be applied to massive data sets.

4.1.1UNIVARIATE HISTOGRAM

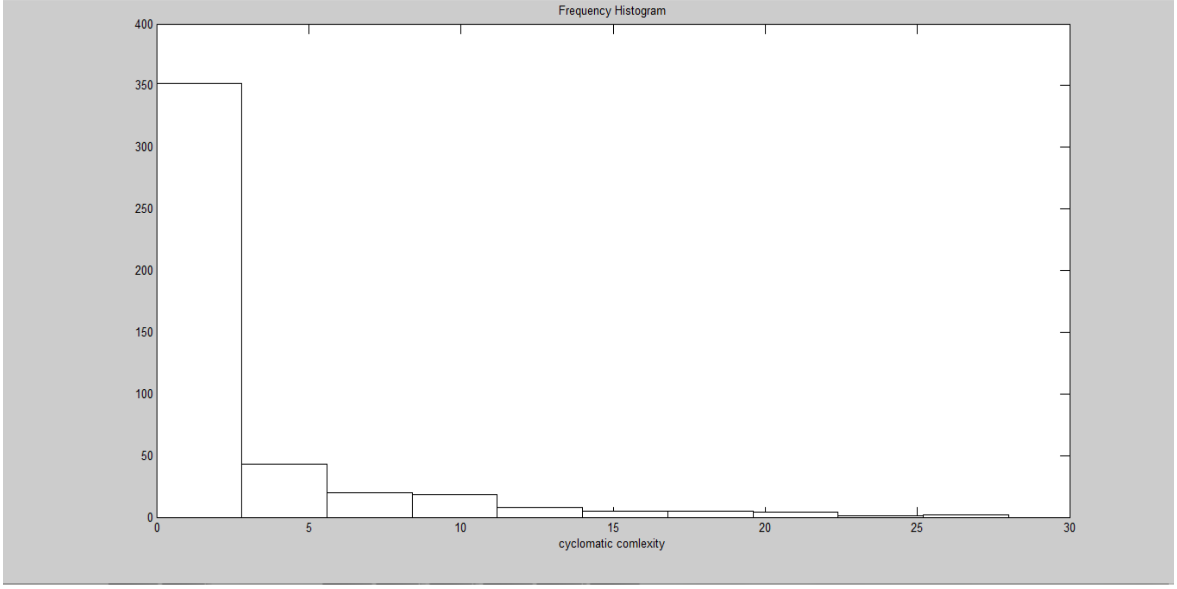
A **frequency histogram** is obtained by first creating a set of bins or intervals that cover the range of the data set. It is important that these bins do not overlap and that they have equal width. We then count the number of observations that fall into each bin. To visualize this information, we place a bar at each bin, where the height of the bar corresponds to the frequency. **Relative frequency histograms** are obtained by mapping the height of the bin to the relative frequency of the observations that fall into the bin. One problem with using a frequency or relative frequency histogram is that they do not represent meaningful probability densities, because the total area represented by the bars does not equal one.

A **density histogram** is a histogram that has been normalized so the area under the curve (where the curve is represented by the heights of the bars) is one. The following algorithm shows the way to plot the relative frequency histogram. The basic MATLAB package has a function for calculating and plotting a univariate frequency histogram called **`hist`**

```
% The 'hist' function can return the
% bin centers and frequencies.
% Use the default number of bins - 10.
[n, x] = hist( Attribute values);
% Plot and use the argument of width = 1
% to get bars that touch.
bar(x,n,1,'w');
title('Frequency Histogram')
xlabel('')
ylabel('')
```

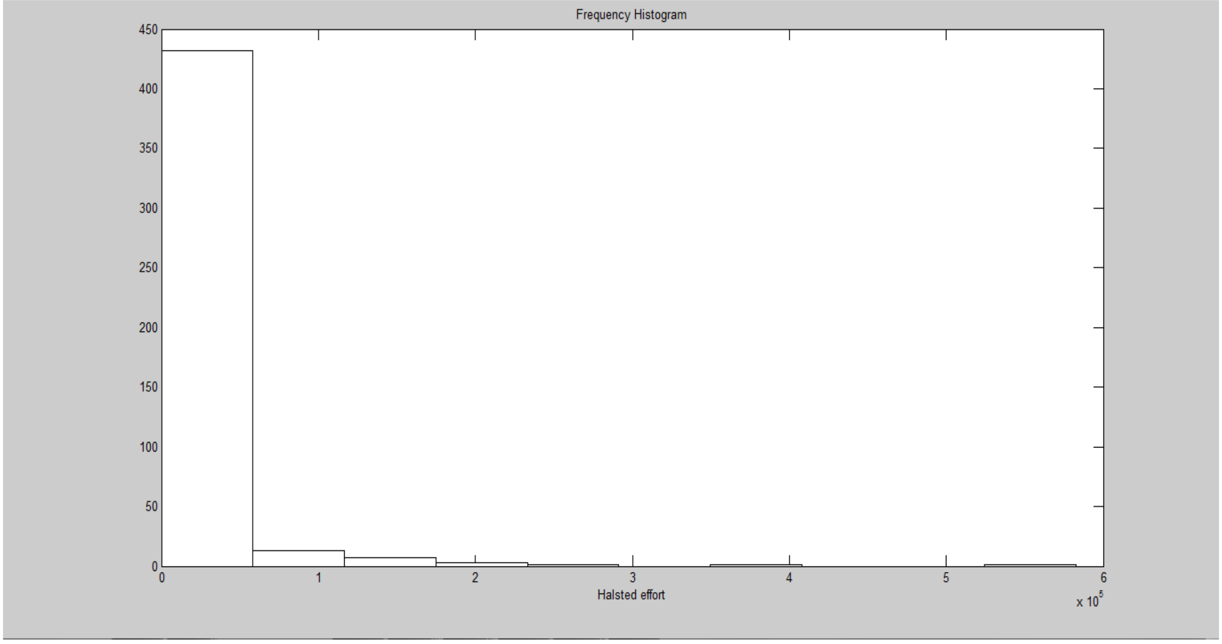
Experiment: Above matlab code was run on the dataset ‘two’ which has 458 instances and 38 attributes. The attributes for which the histogram is plotted are as follows:

- Cyclomatic complexity
- Halsted effort
- Halsted error rate
- LOC_Total

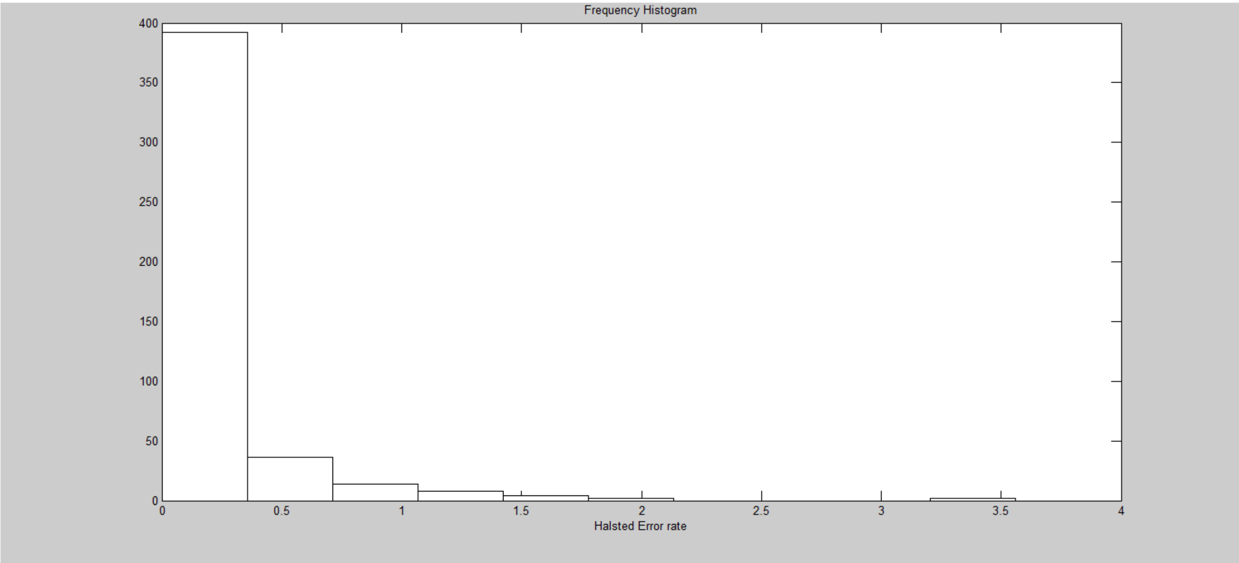


RESULT: The above graph shows the histogram plot for the attribute cyclomatic complexity. Cyclomatic complexity is used to indicate the complexity of a program. It measures the number of linearly independent paths through the program source code. For an efficient program cyclomatic complexity should be less than 10.

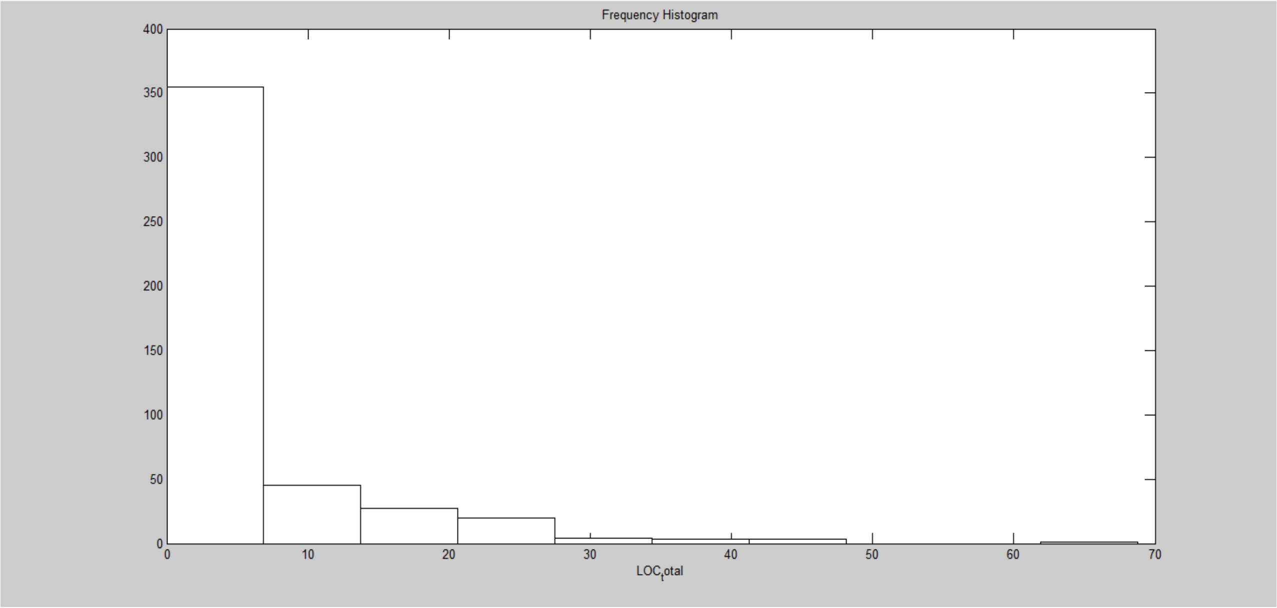
The maximum frequency lies between zero and five. Thus for the maximum number of instances, the cyclomatic complexity is less than 5.



Result: The maximum number of values of the attribute halstead effort lies in the range between 0 and 1.



Result: The maximum number of error rates is less than 0.5.Dataset two has 458 instances out of which around 390 instances have error rate less than 0.5.



Result: In the dataset two around 350 instances have line count of code less than 10.

4.2 BOXPLOTS

Boxplots (sometimes called **box-and-whisker diagrams**) have been in use for many years [Tukey, 1977]. They are an excellent way to visualize summary statistics such as the median, to study the distribution of the data, and to supplement multivariate displays with univariate information. Benjamini [1988] outlines the following characteristics of the boxplot that make them useful:

1. Statistics describing the data are visualized in a way that readily conveys information about the location, spread, skewness, and longtailedness of the sample.

2. The boxplot displays information about the observations in the tails, such as potential outliers.

3. Boxplots can be displayed side-by-side to compare the distribution of several data sets.
4. The boxplot is easy to construct.

5. The boxplot is easily explained to and understood by users of statistics.

Sample **interquartile range** (IQR) is the difference between the first and the third sample quartiles. This gives the range of the middle 50% of the data. It is found from the following

IQR=q(0.75)-q(0.25)

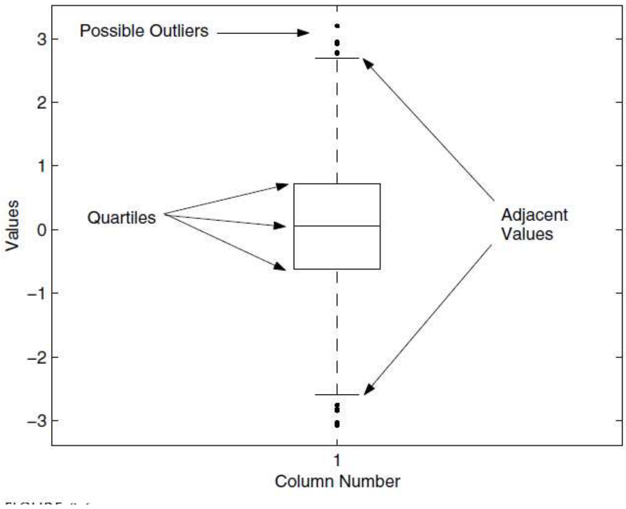
We need to define two more quantities to determine what observations qualify as potential outliers. These limits are the **lower limit** (LL) and the **upper limit** (UL). They are calculated from the IQR as follows

LL=q(0.25)-1.5*IQR

UU=q(0.75)-1.5*IQR

Observations outside these limits are **potential outliers**. In other words, observations smaller than the LL and larger than the UL are flagged as interesting points because they are outlying with respect to the bulk of the data. **Adjacent values** are the most extreme observations in the data set that are within the lower and the upper limits. If there are no potential outliers, then the adjacent values are simply the maximum and the minimum data points.

Following figure shows the example of Box-plot.



4.3 QUANTILE BASED PLOTS

We can use quantile-based plots to visually compare the distributions of two samples. These are also appropriate when we want to compare a known theoretical distribution and a sample. In making the comparisons, we might be interested in knowing how they are shifted relative to each other or to check model assumptions, such as normality. There are several versions of quantile-based plots. These include **probability plots, quantile-quantile plots .**

The probability plot has historically been used to compare sample quantiles with the quantiles from a known theoretical distribution, such as normal, exponential, etc. Typically, a q-q plot is used to determine whether two random samples were generated by the same distribution. The q-q plot can also be used to compare a random sample with a theoretical distribution by generating a sample from the theoretical distribution as the second sample. Finally, we have the quantile plot that conveys information about the sample quantiles.

Given a data set $x1, \ldots, xn$, we order the data from smallest to largest. These are called the **order statistics**, and we denote them on axis.

(1)……….x(n).

The u ($0 < u < 1$) quantile $q(u)$ of a random sample is a value belonging to the range of the data such that a fraction u (approximately) of the data are less than or equal to u. The quantile denoted by $q(0.25)$ is also called the **lower quartile**, where approximately 25% of the data are less than or equal to this number. The quantile $q(0.5)$ is the **median**, and $q(0.75)$ is the **upper quartile**. We need to define a form for the u in quantile $q(u)$. For a random sample of size n, we let u(i)=1-0.5/n.

4.3.1 PROBABILITY PLOT

A probability plot is one where the theoretical quantiles are plotted against the ordered data, i.e., the sample quantiles. The main purpose is to visually determine whether or not the data could have been generated from the given theoretical distribution. If the sample distribution is similar to the theoretical one, then we would expect the relationship to follow an approximate straight line. Departures from a linear relationship are an indication that the distributions are different.

The following MATLAB code is used to display the probability plot.

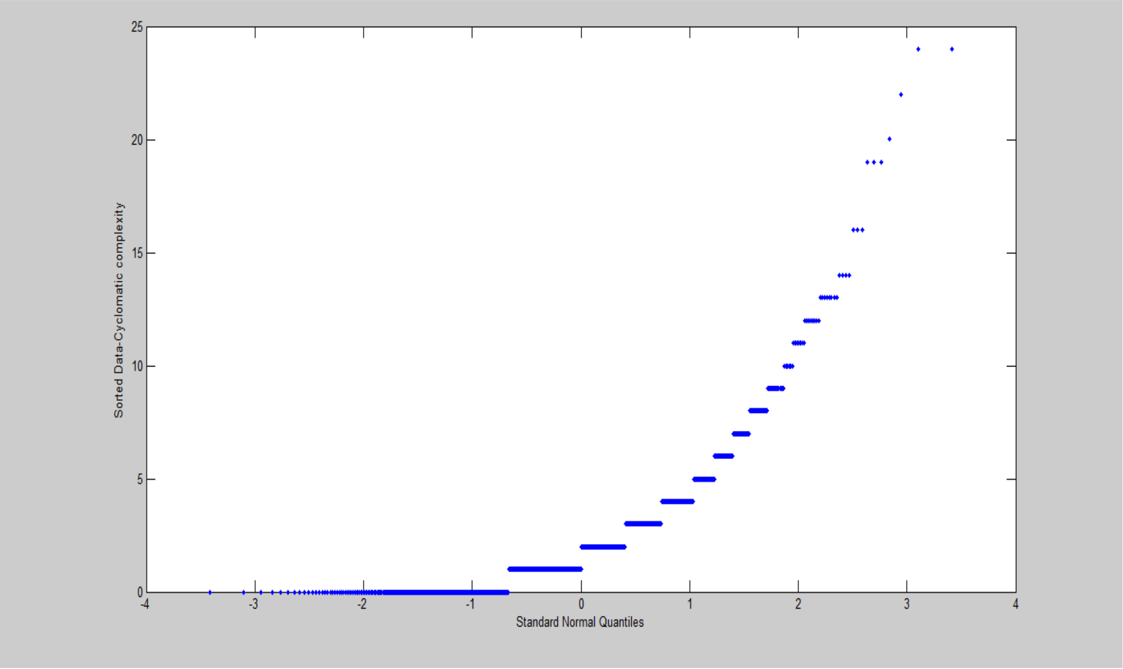
```
x = sort( dataset attribute);
n = length( dataset attribute);
% Get the probabilities.
prob = ((1:n)-0.5)/n;
% Now get the theoretical quantiles for
% a normal distribution.
qp = norminv(prob,0,1);
```

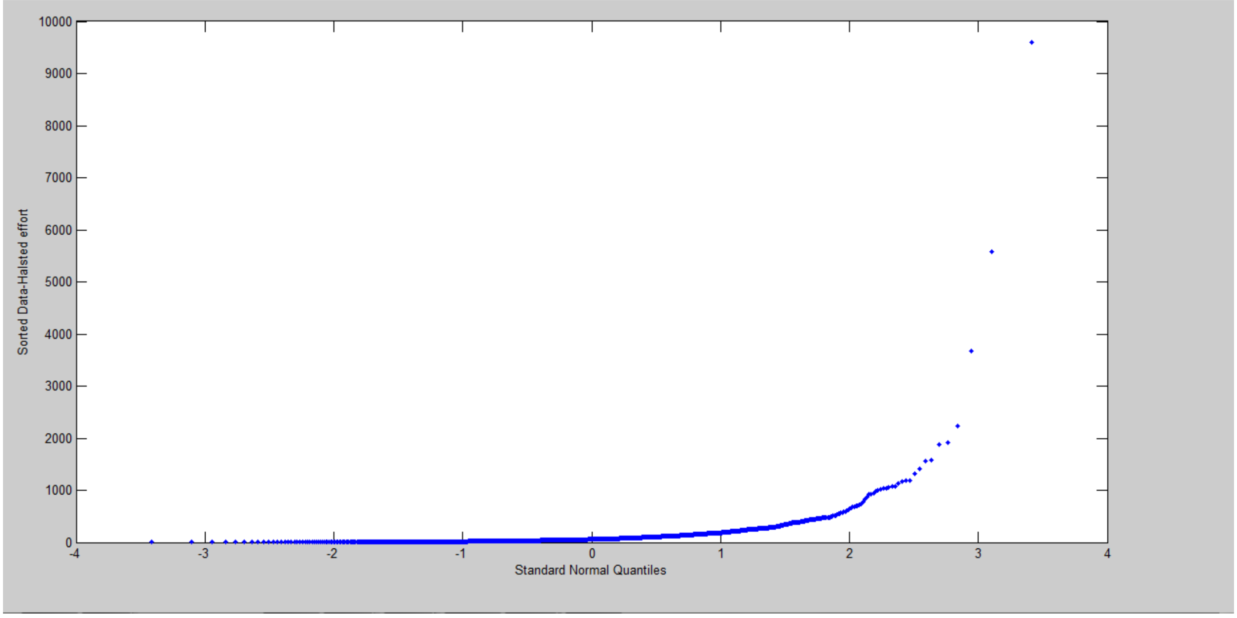
```
% Now plot theoretical quantiles versus
% the sorted data.
plot(qp,x,'.')
ylabel('Sorted Data')
xlabel('Standard Normal Quantiles')
```

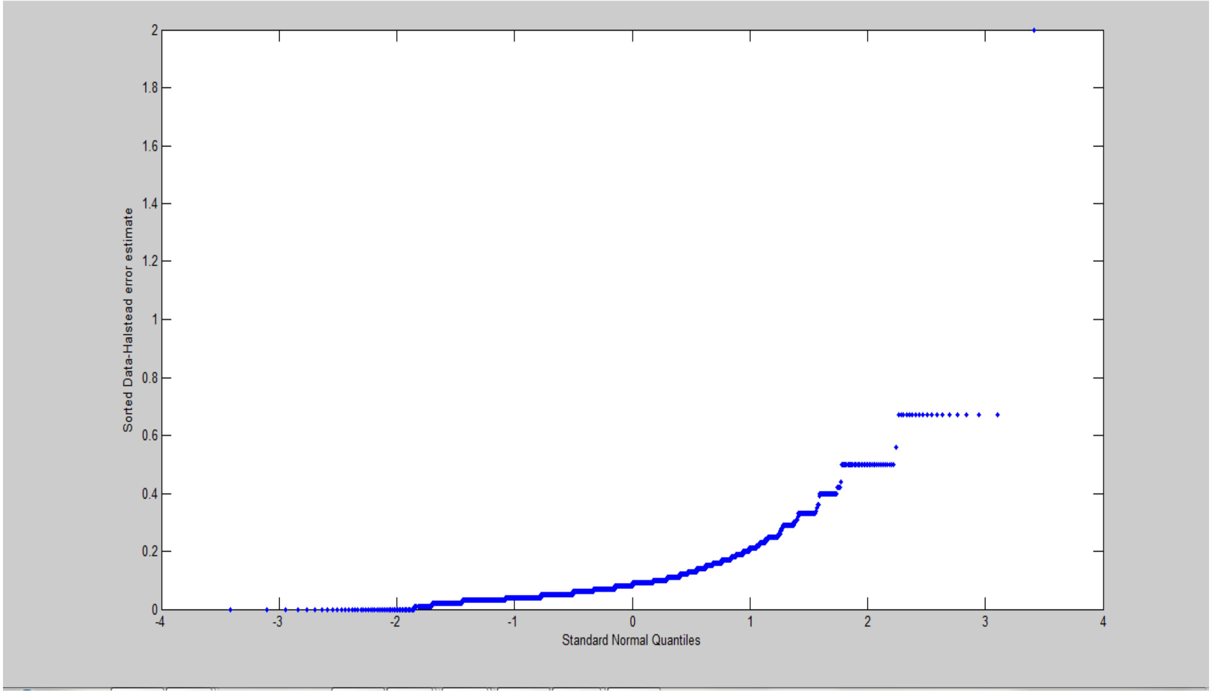
EXPERIMENT:The above MATLAB code was applied on the following datasets.

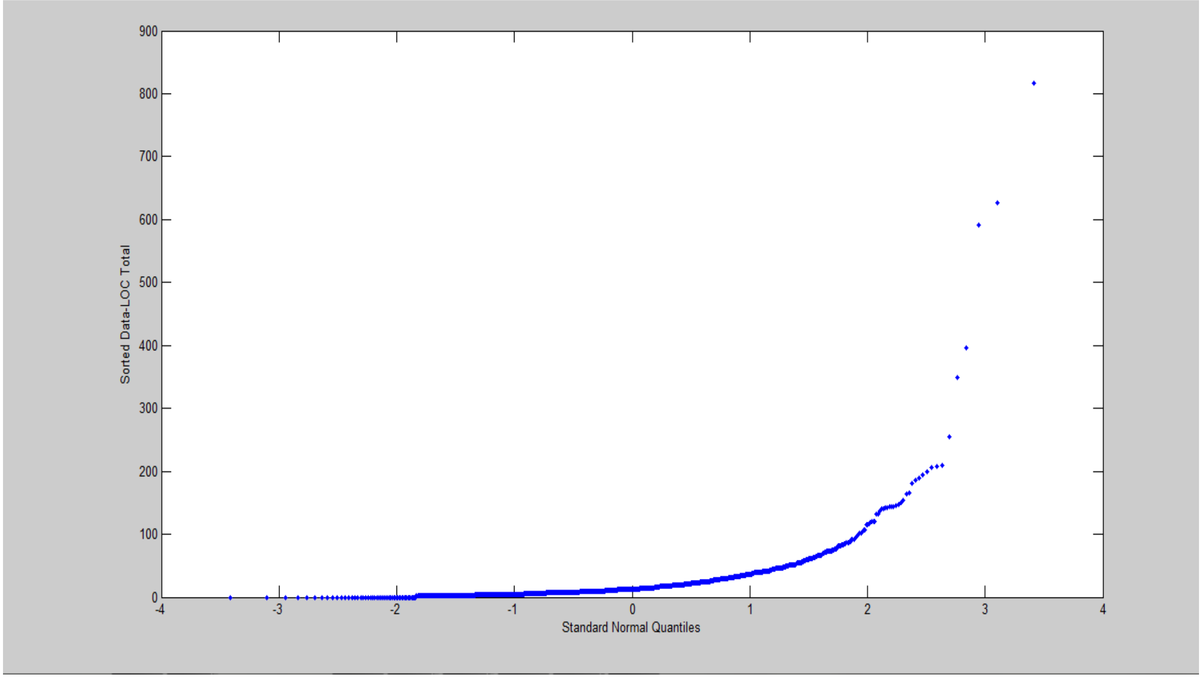
1.Dataset Two: This dataset has 39 attributes.The attributes for which we have plotted the probability plot are as folows:

- Cyclomatic complexity
- Halsted effort
- Halsted error estimate
- Total_LOC




Sorted Data-Halsted effort
Standard Normal Quantiles
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
-4
-3
-2
-1
0
1
2
3
4

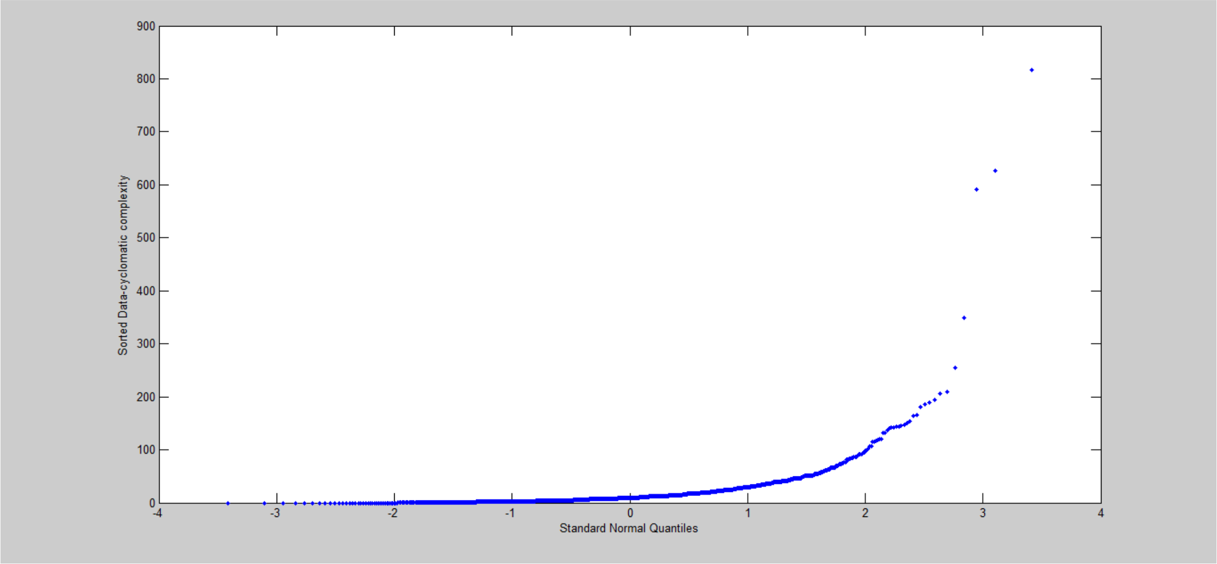

Sorted Data-Halstead error estimate
Standard Normal Quantiles
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-4
-3
-2
-1
0
1
2
3
4



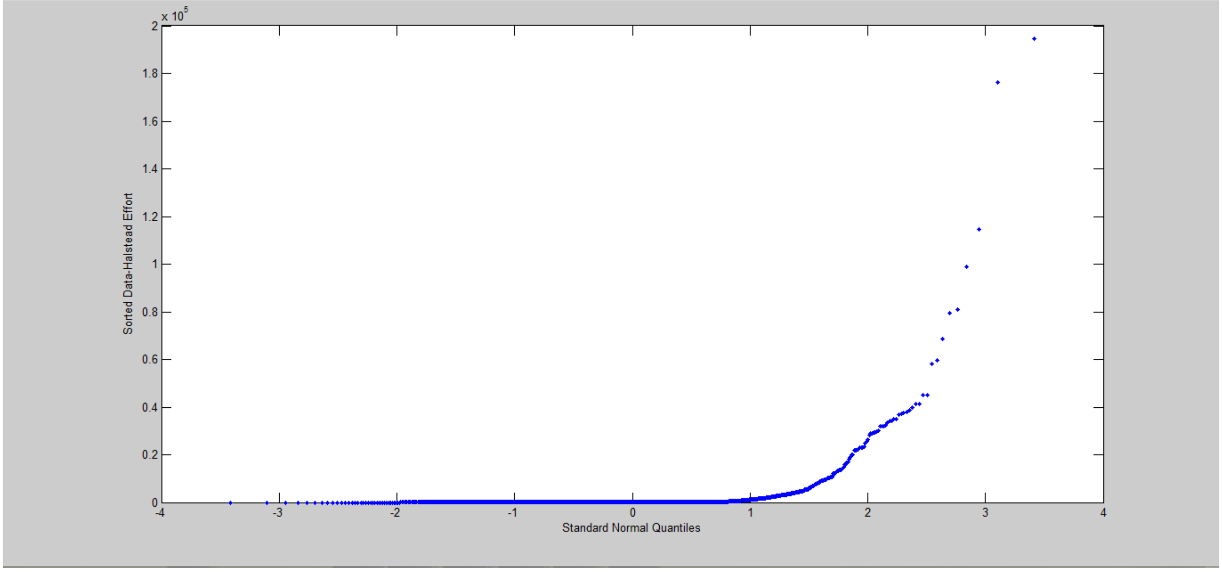
Result: The curvature in the above graphs shows that the distribution shows deviation from the theoretical distribution.Thus we can conclude that the values in the above listed attributes are not normally distributed.

DATASET THREE: This dataset has 403 instances and 38 attributes.We have plotted the probability plot for the same attributes as that of dataset Two**.**

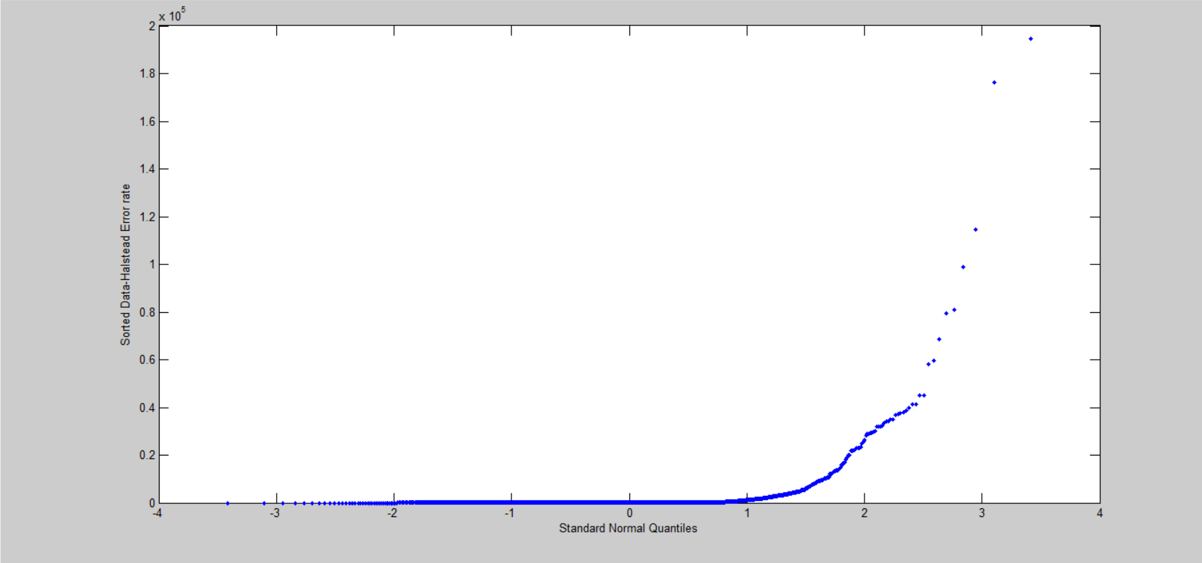
1.Cyclomatic complexity



2. Halstead Effort



3. Halstead Error rate



Result: The curvature in the above graphs show deviation from the theoretical distribution. Thus the above attributes for which we have plotted the probability plot are not normally distributed.

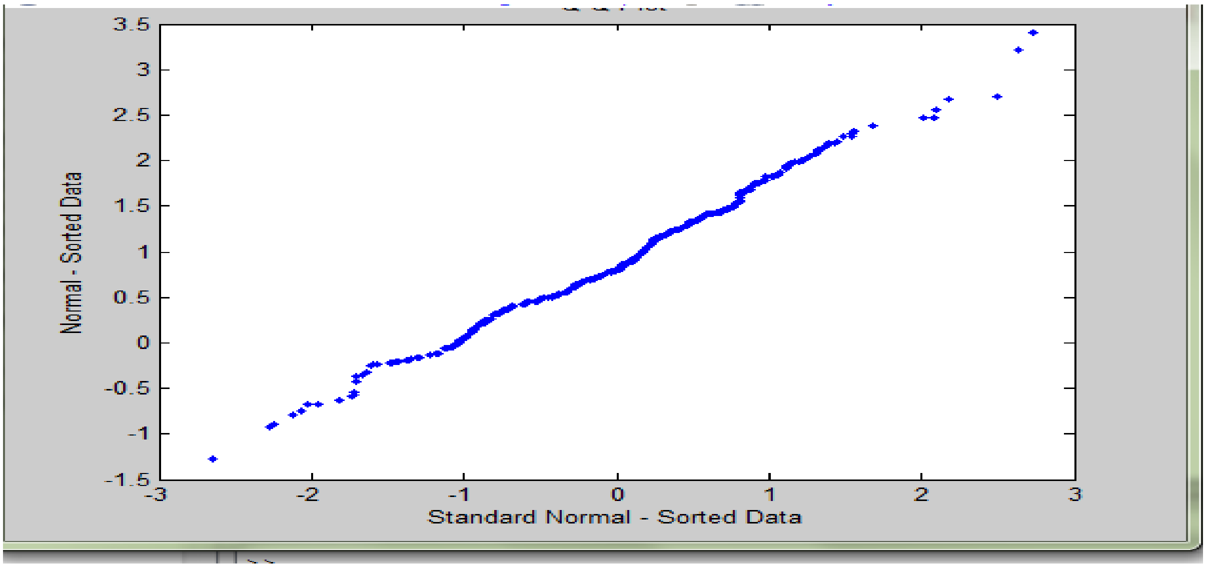
4.3.2 QUANTILE-QUANTILE PLOT

The q-q plot was originally proposed by Wilk and Gnanadesikan [1968] to visually compare two distributions by graphing the quantiles of one versus the quantiles of the other. Either or both of these distributions may be empirical or theoretical. Thus, the probability plot is a special case of the q-q plot. A q-q plot is a plot of the quantiles of the first data set against the quantiles of the second data set. By a quantile, we mean the point below which a given fraction (or percent) of points lies. That is, the 0.3 (or 30%) quantile is the point at which 30% percent of the data fall below and 70% fall above that value.

A 45-degree reference line is also plotted. If the two sets come from a population with the same distribution, the points should fall approximately along this reference line. The greater the departure from this reference line, the greater the evidence for the conclusion that the two data sets have come from populations with different distributions.

Following MATLAB code shows the q-q plot for random data values .

```
% Generate the samples - same size.
x = randn(1,300);
% Make the next one a different mean
% and standard deviation.
y = randn(1,300)*.75 + 1;
% Find the order statistics - sort them.
xs = sort(x);
ys = sort(y);
% Construct the q-q plot - do a scatterplot.
plot(xs, ys, '.')
xlabel('Standard Normal - Sorted Data')
ylabel('Normal - Sorted Data')
```



5. SCATTER PLOT

The scatterplot is a visualization technique that enjoys widespread use in data analysis and is a powerful way to convey information about the relationship between two variables. To construct one of these plots in 2-D, we simply plot the individual (*xi* , *yi*) pairs as points or some other symbol. For 3-D scatterplots, we add the third dimension and plot the (*xi* , *yi* , *zi*) triplets as points.

The MATLAB function scatter is used to get the 2-D scatter plot of the data. Scatter(X,Y,S,C,M) where **X** and **Y** are the data vectors to be plotted, and the other arguments areoptional. **S** can be either a scalar or a vector indicating the area (in units of points-squared) of each marker. **M** is an alternative marker (default is the circle), and **C** is a vector of colors. The scatterplot is a visualization technique that enjoys widespread use in data analysis and is a powerful way to convey information about the relationship between two variables. To find the relationship between the two variables we can take any two variables from the dataset. For example in the following algorithm attributes belonging to column 8 and nine of a particular dataset have been taken.

```
scatter(Dataset(:,8),Dataset(:,9))
xlabel(labcol{x})
ylabel(labcol{y})
scatter(X,Y,S,C,M)
% If we want to use different colors for the groups,
% we can use the following syntax. Note that this
% is not the only way to do the colors.
ind0 = find(midden==0); % Red
ind1 = find(midden==1); % Green
ind2 = find(midden==2); % Blue
% This creates an RGB - 3 column colormap matrix.
C = zeros(length(midden),3);
C(ind0,1) = 1;
C(ind1,2) = 1;
C(ind2,3) = 1;
```

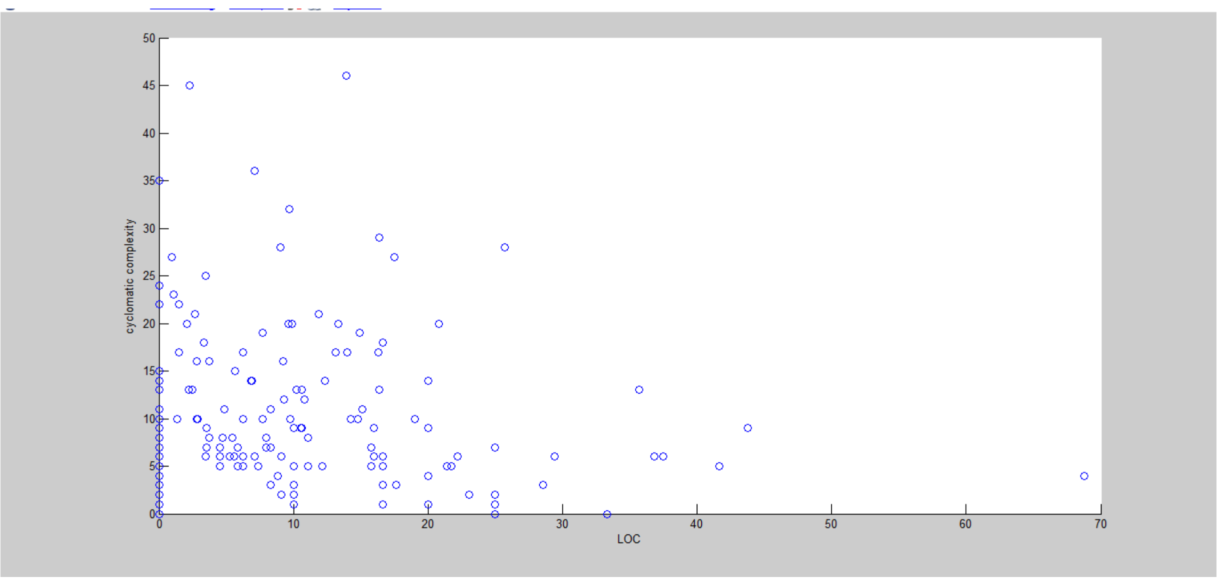
EXPERIMENT: The above MATLAB code was applied on the following datasets

DATASET: TWO

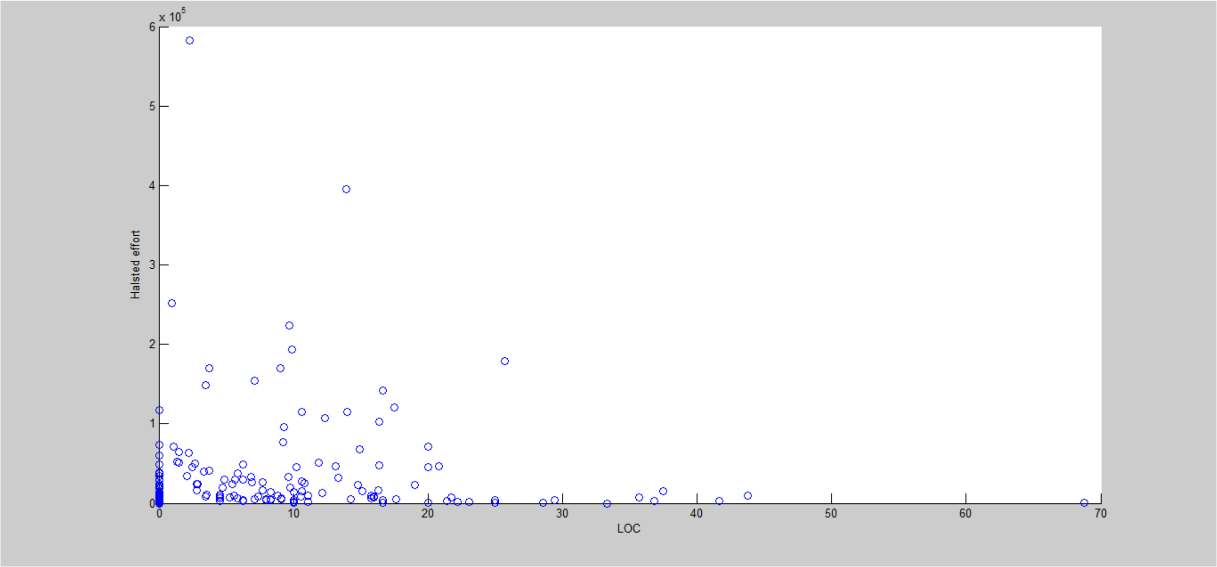
Following figures show the scatter plot between the following attributes of the

Dataset’ two’.

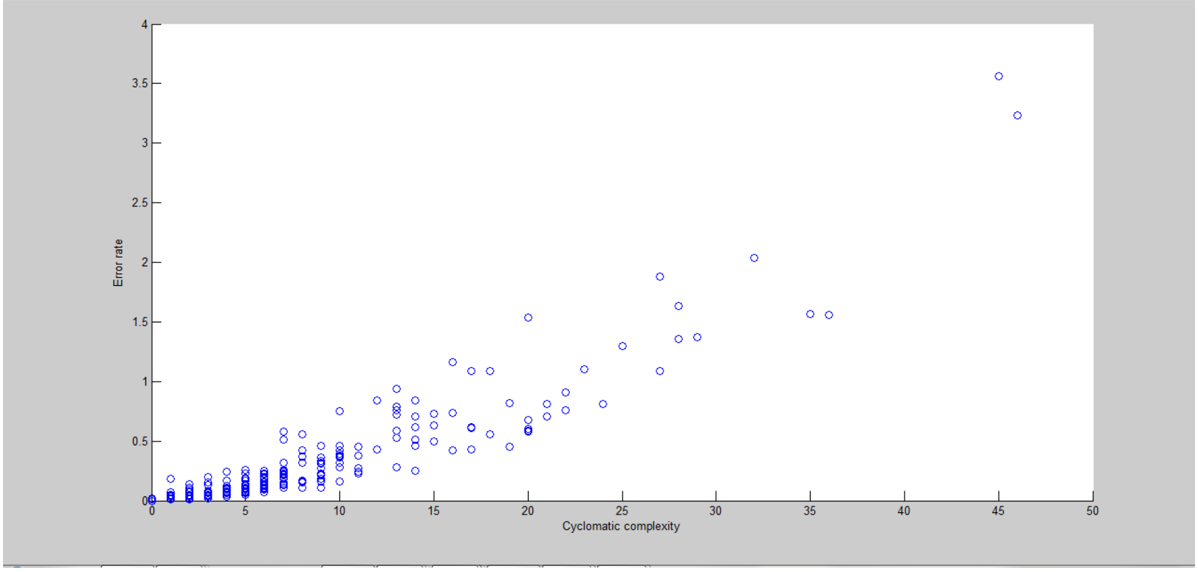
- LOC vs. Cyclomatic complexity
- LOC vs Halsted effort
- LOC vs error rate
- Cyclomatic complexity vs error rate



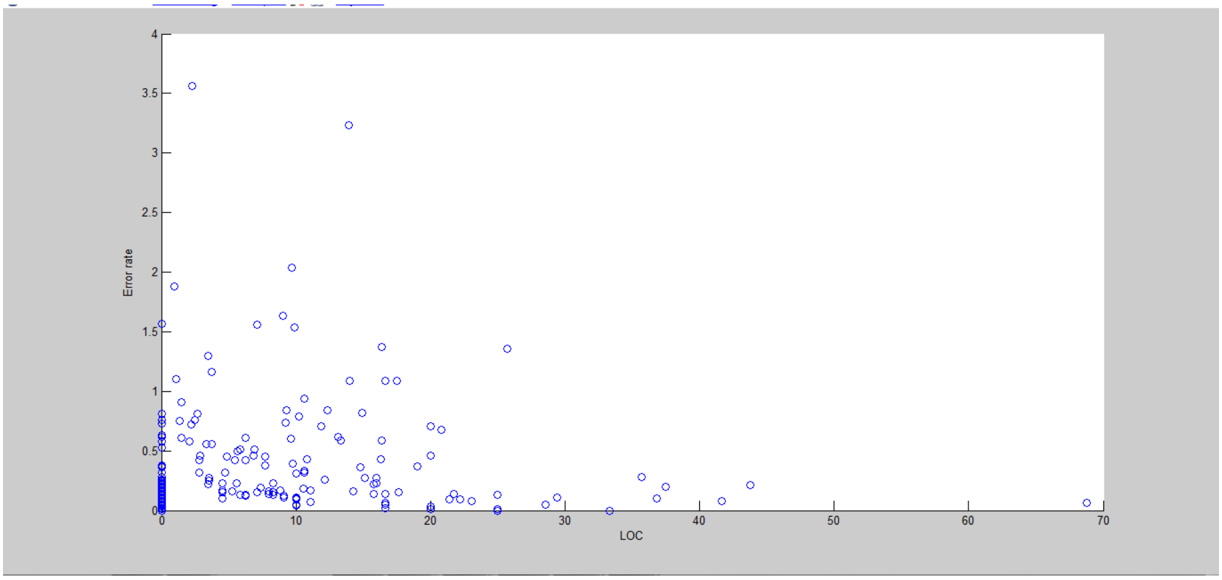
Result: From the above scatter plot shows that for a single value of x there are different values of y.For example for x=0,the value of y ranges from 0 to 25.Thus we can conclude that line count of code does not depend upon the cyclomatic complexity.



Result: There is no proper relationship between the attributes line count of code and halstead effort.



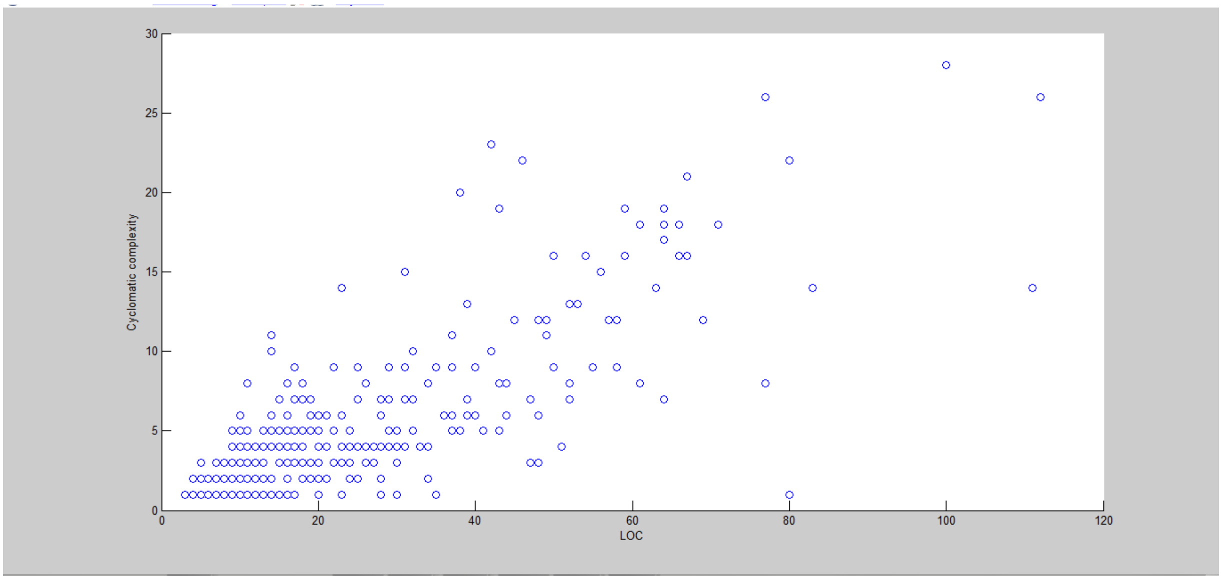
Result: The relationship can be considered linear to some extent, as we increase the value in x axis the corresponding values in the y axis also increases. But there are some potential outliers due to which the relationship becomes non linear.



Result: The above scatter plot shows that there is no relationship between Line count of code and error rate**.**

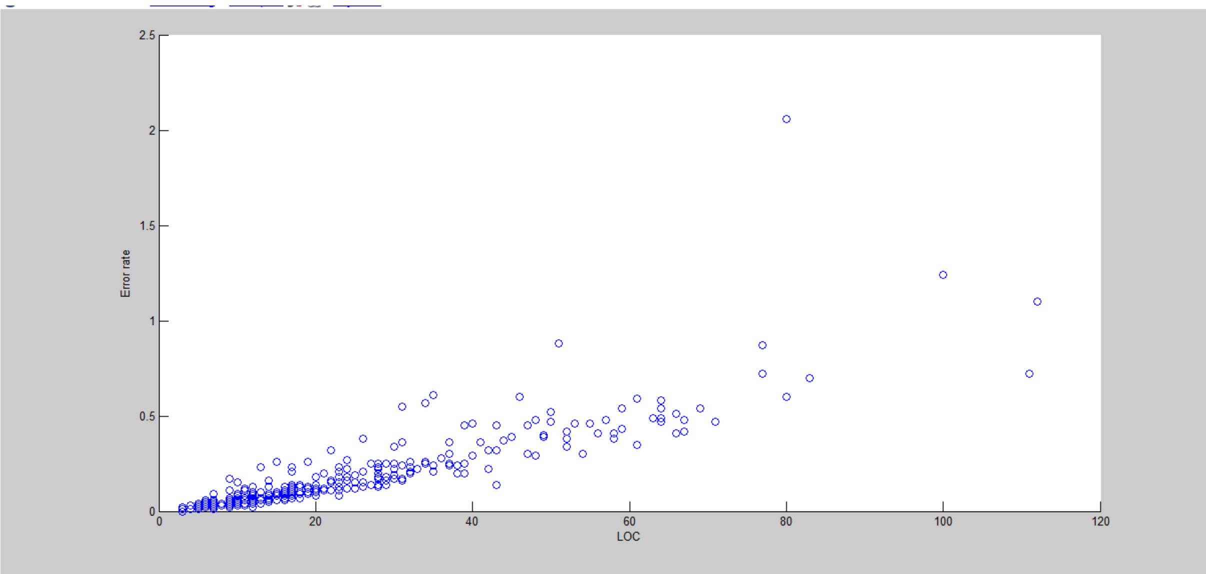
Dataset Three: Dataset three has 38 attributes. To plot the scatter plot we have taken the same pair of attributes as that for dataset two.

1. LOC vs. Cyclomatic complexity



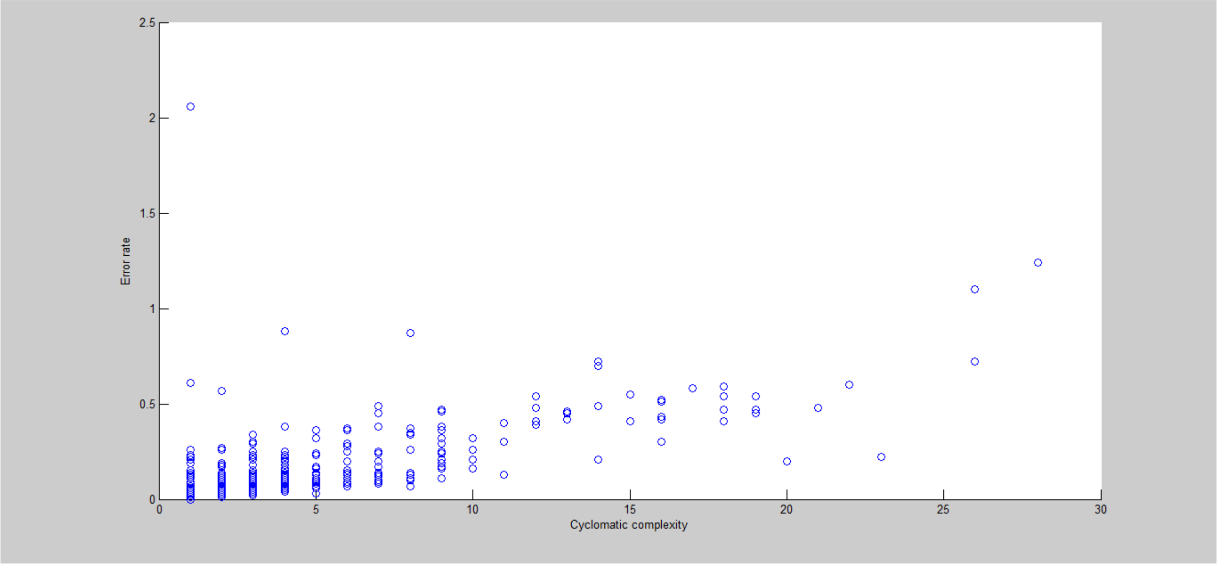
Result: There is a basic linear relationship between line count of code and cyclomatic complexity for y<=10.As the values on the y axis increases the relationship becomes non-linear.

2. LOC vs. Error rate



Result: The relationship between line count of code and error rate can be considered to be linear to some extent.i.e as the line count of code increases error rate also increases. But there are some outliers also which counterfeit the above result.

3. Cyclomatic complexity vs. Error rate



Result: The above scatter plot between error rate and cyclomatic complexity denotes that variation in error rate does not depend upon the cyclomatic complexity of the code. For example if the cyclomatic complexity is five then the error rate lies in between 0 and 0.5.

6. SUMMARY AND CONCLUSION

EDA through its various graphical and non-graphical techniques provides convenient way to analyze the data and find out various relationships between the variables. In practical applications the *s*electio*n* and preprocessin*g* of the data may be even more important than the choice of the analysis method. For example principal component analysis helps us to reduce the dimensionality of the data without reducing the variability. Probability plot determines whether the data is normally distributed or not similarly scatter plot reveals the relationship between the two variables. While analyzing the data it is very necessary to find out the dependency of one attribute on other so that we can develop effective analytical models.

In datasets sometimes we have outliers i.e. the values which are out of range of the distribution and should be eliminated. Techniques like box plot easily help us to find out such outliers and they also help us to find out the extreme values.

Similarly there are some other EDA techniques also which helps in preprocessing of the data so that we can get effective results while developing suitable analytical models.

REFERENCES:

[1] Wendy L. and Angel R. Martinez ‘Exploratory Data Analysis with MATLAB’. International Standard Book Number 1-58488-366-9.October 2004.

[2]: http://en.wikipedia.org/wiki/Exploratory_data_analysis. Accessed on : 11th May 2012.

[3]: http://www.itl.nist.gov/div898/handbook/eda/eda.htm. Accessed on: 20th May 2012

[4]: Lindsay I Smith.’A Tutorial on Principal Component Analysis’. Feb 26 2002.