CS252 Project Report: Comparing the Intricacies R versus Octave

Suraj Natarajan, Shruti A Sharma
San José State University

December 13, 2014

Abstract

Octave and R are two interactive, highlevel programming languages used in scientific computing. The languages have a lot in common but have very different target audiences and focus. R is primarily used by the statistical community for advanced data analysis and research in statistical methodology. Octave is primarily used by engineers. As a part of this project we aim to highlight the differences and commonalities between the two languages.

1 Introduction

As a part of this project, we have used R and Octave to analyse and derive statistical analysis of Disabled and Non workers in India. We have also plotted several statistics using the R studio and Octave and found interesting facts. Mentioned below are the details of the dataset that we have used.

Dataset Description : The dataset is in the form of a CSV file and has several features as listed below

Features	Description
State, District Code, Area Name	Regions of the non workers and disabled people
Rural, Urban	If the people live in Rural or Urban areas
Activity of Non-Worker	What do the Non-Workers do
Total Number of disabled Non-workers	Count of the disabled non-workers
Number of disabled M/ F Non-workers	Count of the disabled M/F non-workers
Numbers based on Disability	Count of the different disabled non-workers

The current population of India is 1.27 billion(18 percent of world population), the second largest country in terms of the human resources in the world. But it is also the home to 26 million people with disability.There are 29 states and 7 union territories in India. In this project we will analyze non-workers based on the type of disability in India in each state and union territory.As per the census 2011 the total count of people who are disabled non-workers is about 17 million. The various categories of disabled non-workers are Student,

Household duties, Dependent, Pensioner, Rentier, Beggars, Others. We will compute the statistics of disabled non-workers both in rural and urban India as well as statistics for each state of India. Basically we will be writing the necessary queries for computational so as to make the current scenario more vivid for considering the disabled persons for employment.

Source for the dataset : http://data.gov.in/catalogs/?query= – This site is controlled by the Indian Government. This data set has the data that was maintained from 1947 to 2011(latest census) by government of India.

2 Analysis using R and Octave

As a part of this project we have used the intricate features that R and Octave provides us in order to analyse and plot the statistics of non workers and disabled. All the R files are stored using the .r extension and all the Octave files are stored similar to Matlab with a .m extension.

2.1 Loading Data in R and Octave

R

Listed below are the commands used to read the CSV file into R Commands used :

```
setwd("I:\\SJSUSem2\\CS252\\Project")
Non_Workers <- read.csv(file="Disabled-Non-Workers.csv",head=TRUE,sep=",")
```

The setwd command enables us to set the path from where the CSV file has to be loaded and the read.csv command will help to read the CSV file and load it into the Non Workers. The values are comma separated and that is shown using the last argument of read.csv file.

Octave

Listed below are the commands used to read the CSV file into Octave Commands used :

```
   fid = fopen('Disabled-Non-Workers.csv', 'r');
   Non_Workers = textscan(fid,'%s%d%d%s%s%s%d%d%d%d%d%d%d%d%d%d
%d%d%d%d%d%d%d%d%d%d%d%d%d%d%d','delimiter',',');
```

The fopen command opens the CSV file in read mode and the textscan scans the data and stores the array into Non Workers.

2.2 Data Processing in R and Octave

R

Some of the character fields of the dataset are in uppercase and some are in lowercase. In order to create all the fields to be of the same format, we can use the toupper() function to convert all the characters to Uppercase.
Commands used :

```
area_name <-  toupper(area_name)
```

Octave
Some of the character fields of the dataset are in uppercase and some are in lowercase. In order to create all the fields to be of the same format, we can use the tolower() function to convert all the characters to lower case. We have used different functions in R and Octave just to show the syntactic differences.

```
area_name = tolower(area_name);
```

2.3 Reading and Writing to Console in R and Octave

R
Scan : is used to read data into a vector or list from the console or file. It has the arguments What (the type of what gives the type of data to be read) and nmax(the maximum number of data values to be read) Print : Prints its argument onto the console

Commands used :

```
print("Enter your choice\n");
choice <-  scan(what=character(),nmax=1)
```

Octave

The disp function is used to display text on the console. The input function is used for an interactive dialog with a user.

Commands used :

```
disp("a.Analysis based on Grand Total b.Analysis based on Gender\n");
choice= input('Enter your choice: ','s');
```

2.4 Creating vectors and Matrices in R and Octave

R
In R everything is a vector. Vectorized functions: process whole vectors

```
This means that, in R, typing 6 tells R something like
<start vector, type=numeric, length=1>6<end vector>
```

We can create vectors in R using the command v = c (data1, data2...). This will create a row vector which can further be used for data processing and analysis.

```
Example : Suppose B and C are vectors
Instead of explicit element-by-element loop
for (i in 1:N) { A[i] < B[i] + C[i] }
invoke the implicit elem.-by-elem.
Operation: A <- B + C
```

Commands used in our project :

```
dflist <- c("INDIA","ANDAMAN", "ANDHRA)
```

matrix creates a matrix from the given set of values. as.matrix attempts to turn its argument into a matrix. is.matrix tests if its argument is a (strict) matrix.

Commands used :

```
disabled_gen <- matrix(c(newdata_gen$male[1:1],newdata_gen$Female[1:1],
newdata_gen$male[2:2],newdata_gen$Female[2:2],
newdata_gen$male[3:3],newdata_gen$Female[3:3],
newdata_gen$male[4:4],newdata_gen$Female[4:4],
newdata_gen$male[5:5],newdata_gen$Female[5:5],
newdata_gen$male[6:6],newdata_gen$Female[6:6],
newdata_gen$male[7:7],newdata_gen$Female[7:7]),ncol=7,byrow=FALSE)
disabled_gen <- as.table(disabled_gen)
```

Octave

We can create vectors in Octave using the command v = [data1, data2...]. This will create a row vector which can further be used for data processing and analysis.

```
Example:
Here is how we specify a row vector in Octave:

octave:1> x = [1, 3, 2]
x = 1  3  2
Note that the vector is enclosed in square brackets;
each entry is separated by an optional comma. x = [1 3 2] results in the same row vector.
To specify a column vector, we simply replace the commas with semicolons:

octave:2> x = [1; 3; 2]
x =

  1
  3
  2
From this you can see that we use a comma to go to the next column of a vector (or matrix)
and a semicolon to go to the next row. So, to specify a matrix, type in the rows
(separating each entry with a comma) and use a semicolon to go to the next row.

octave:3> A = [1, 1, 2; 3, 5, 8; 13, 21, 34]
A =

   1   1   2
   3   5   8
  13  21  34
```

Commands used :

```
types = [Non_Workers{8}, Non_Workers{9}];
```

Matrices can be created in a similar way in Octave using the command [data1 data2 ; data3 data4]

Commands used :

```
types_data1 = [types(pos,1) types(pos,2); types(pos+1,1) types(pos+1,2)]
```

2.5 Using Subsets of data

R

We can use the Subset command which returns subsets of vectors, matrices or data frames which meet certain conditions. This helps us in working on specific data which we want to analyse.

Commands used :

```
newdata_urban <- subset(Non_Workers, Area.Name==area_name &
Total..Rural..Urban == "Urban" & Activity.of.Non.worker != "Total",
select=Activity.of.Non.worker:Total.disabled.non.worker...Persons)
```

Octave

There is no specific subset function in Octave. The subset of columns can be extracted using the command listed below which is the same as the commad to create Vectors.

Commands used :

```
types = [Non_Workers{8}, Non_Workers{9}];
```

2.6 Storing data tables

A data frame is used for storing data tables. It is a list of vectors of equal length.

Commands used :

```
Student <- c(newdata_gen_rural$male[1:1],newdata_gen_rural$Female[1:1],
newdata_gen_urban$male[1:1],newdata_gen_urban$Female[1:1])
Household_duties <-c(newdata_gen_rural$male[2:2],newdata_gen_rural$Female[2:2],
newdata_gen_urban$male[2:2],newdata_gen_urban$Female[2:2])
dat <- data.frame(Student, Household_duties)
```

2.7 Plotting Bar Graph using R and Octave

R

A bar graph of a qualitative data sample consists of vertical parallel bars that shows the frequency distribution graphically.

Commands used :

```
names(newdata)[names(newdata)=="Total.disabled.non.worker...Persons"] <- "two"
bp<-barplot(newdata$two, las=2, names.arg= newdata$Activity.of.Non.worker,
main=paste('Activity of Non-Worker in', area_name), axes = FALSE, col="Green")
options("scipen"=100)
par(mfrow=c(1,1))
usr <- par("usr")
par(usr=c(usr[1:2], 0, 800000))
axis(side = 2, at = seq(0, 800000,100000))
text(bp, 0, round(newdata$two, 1),cex=0.6,pos=3)
```

Octave

```
bar (x, y)
bar (y)
bar (x, y, w)
bar (x, y, w, style)
h = bar (. . . , prop, val)
```

Produce a bar graph from two vectors of x-y data. If only one argument is given, y, it is taken as a vector of y-values and the x coordinates are taken to be the indices of the elements. The default width of 0.8 for the bars can be changed using w. If y is a matrix, then each column of y is taken to be a separate bar graph plotted on the same graph. By default the columns are plotted side-by-side. This behavior can be changed by the style argument, which can take the values "grouped" (the default), or "stacked".

Commands used in our project:

```
      h=bar(types_data1,"stacked");
      h=get (gcf, "currentaxes");
      set(h,"fontweight","bold");
      set(h,"xtick",[1 2 3 4 5 6 7]);
      set(h,"xticklabel",['Student';'Household' ;'Dependent';
'Pensioner';'Rentier' ;'Beggar';'Others']);
      title (cstrcat("Activity of Non-Worker based on Disability
 in Rural part of ", " ", area_name));
legend('Seeing', 'Hearing', 'Speech', 'Movement', 'Mental retardation',
'Mental illness', 'Any other', 'Multiple disability')
```

2.8 Plotting Pie Chart using R and Octave

R

Pie charts are created with the function pie(x, labels=) where x is a non-negative numeric vector indicating the area of each slice and labels= notes a character vector of names for the slices. Commands used :

```
slices <- c(newdata$two)
lbls <- c("Student", "Household duties", "Dependent", "Pensioner",
```

```
"Rentier", "Beggar  Vagrants etc.", "Others")
pct <- round(slices/sum(slices)*100, digits = 2)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,labels = lbls, col=rainbow(length(lbls)),
main=paste('Activity of Non-Worker in', area_name))
```

Octave

```
pie (x)
pie (x, explode)
pie (. . . , labels)
pie (h, . . . );
h = pie (. . . );
```

Produce a 2-D pie chart. Called with a single vector argument, produces a pie chart of the elements in x, with the size of the slice determined by percentage size of the values of x. The variable explode is a vector of the same length as x that if non zero explodes the slice from the pie chart. If given labels is a cell array of strings of the same length as x, giving the labels of each of the slices of the pie chart. The optional return value h is a list of handles to the patch and text objects generating the plot.

Commands used in our project:

```
pie ([types(pos,1), types(pos+1,1), types(pos+2,1)], [0, 0, 1]);
title('Pie Chart of Six Larges Oil Nations');
```

2.9 Conditionals and Loops in R

If and For are the extensively used constructs in R

If Commands used :

```
if(flag == 0){
print("Invalid area name, please execute again\n");
quit("default", 0, TRUE)
}
```

For Commands used :

```
for (i in dflist) {
if(area_name == i){
flag = 1;
break;
}
}
```

3 Why use R over Octave

3.1 Pros of R

First, R is a language that was designed specifically for statistical computing and graphics. As a result, it has a wide following in the statistics community, which has in turn created a very extensive set of packages for ML and statistics (see the packages listed on CRAN). The R command line is also very useful when doing interactive plotting and exploration of a dataset. I find that my R programs are a bit shorter than the equivalent code in python.

It is extremely accurate and fast as well (which matters for Big Data)

If the data analysis tasks require standalone computing or analysis on individual servers, R would be helpful.

3.2 Calculations in R

R:

```
x = 2
y = 3
x + y
```

3.3 Vectors and Matrices in R

R:
A vector is a list of numbers that we can do math with. The numbers in the vector are indexed, so that we can access them. Note that vector indexing in R starts with 1 not with zero. So R counts the first element in a vector as element 1, the second as element 2, etc. So if we try to recall an element at the zero position, then R will throw an error. Here are some examples. Vectors x y year and names are assigned as follows

```
x = c(1, 3, 4, 9)
y = c(9.2, 0.3, 3.2, 2.8)
year = seq(2000, 2003, by = 1)  # seq(from, to, by = stepsize)
somev = seq(2000, 2003, length = 10)  # seq(from, to, length = nr. of steps)
names = c("Tom", "Dick", "Harry", "Patrick")
x[0]
```

3.4 Math in R

R:
In R we can add, subtract, multiply, and divide vectors element-by-element.

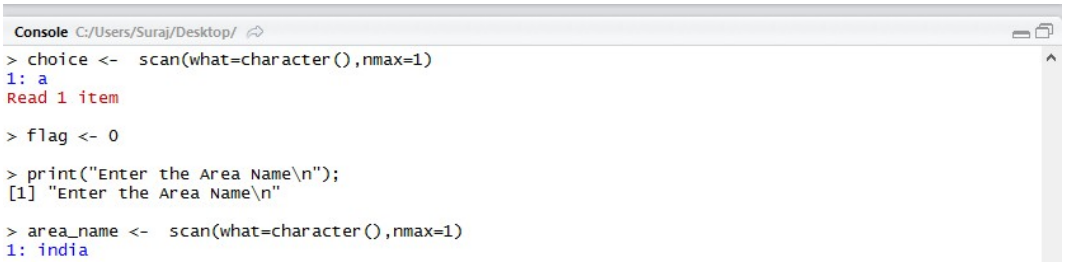
Again define two vectors as

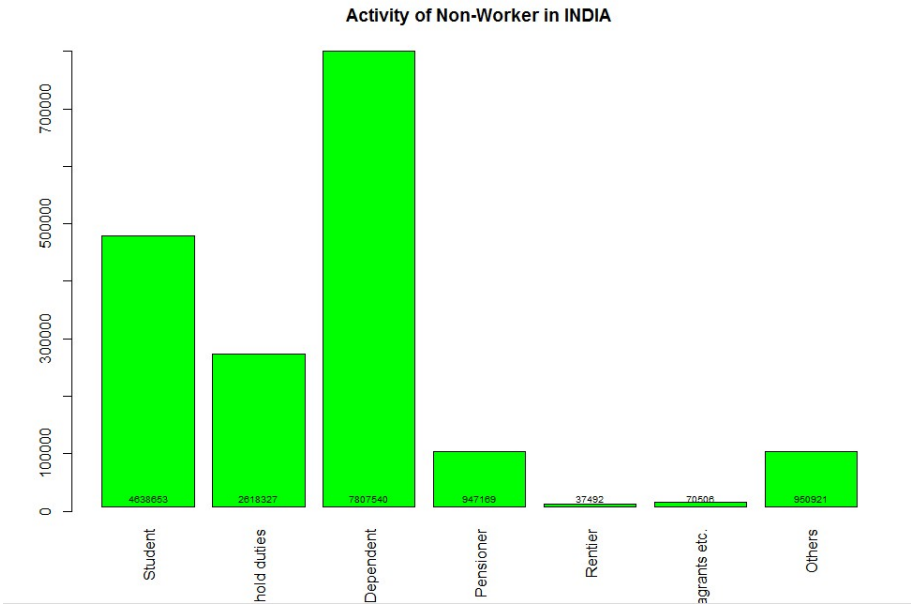
```
x1 = c(1, 3, 4, 9)
x2 = c(2, 5, 6, 3)
x1 + x2
```

3.5 Test Results for Analysis of total number of disabilities in India

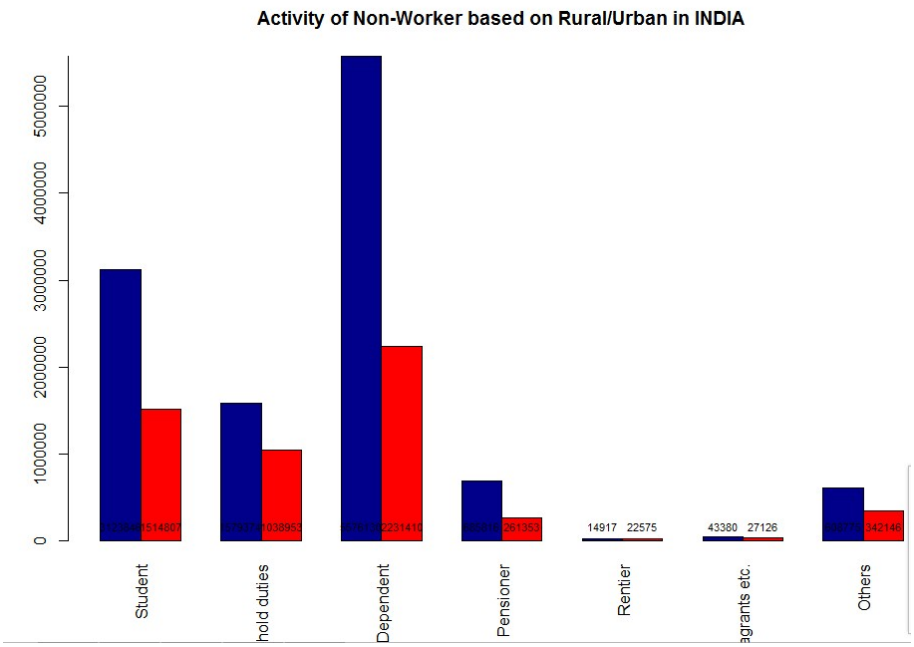
Below is the screen shot of the execution of the test run. We chose choice as character 'a' and area_name as India

R

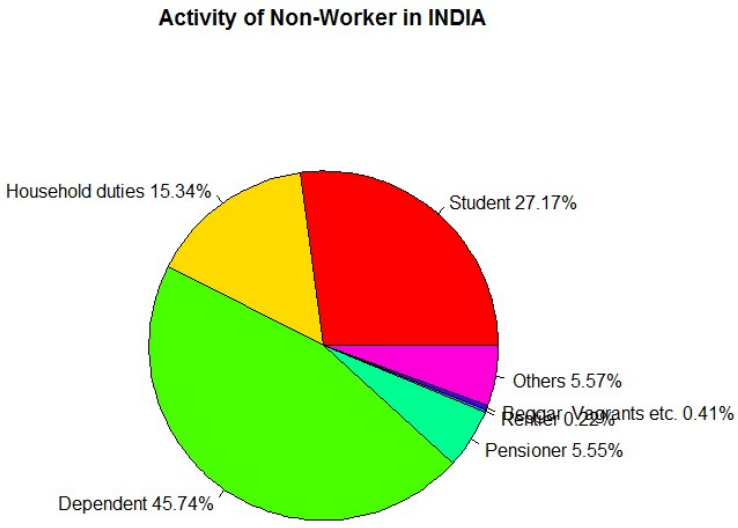




So from the above result it is clear that Activity of Disabled Non-workers in India is mostly Dependent.

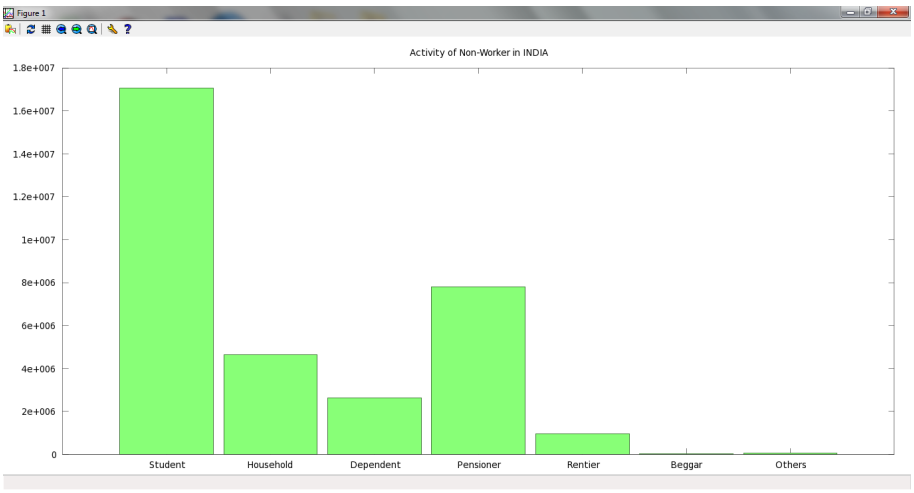


Comparision of the Non-workers in India in Rural and Urban areas.



Percentage of the summary of the non-disabled Non-workers in India. It can be seen the major chunk of the activity of non-worker is Dependent, Student and Household duties.

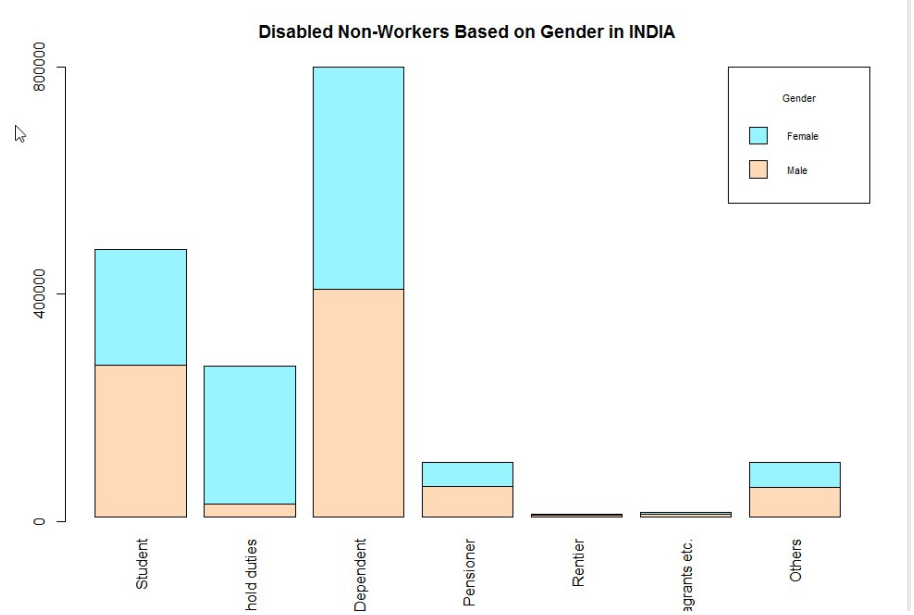
Octave



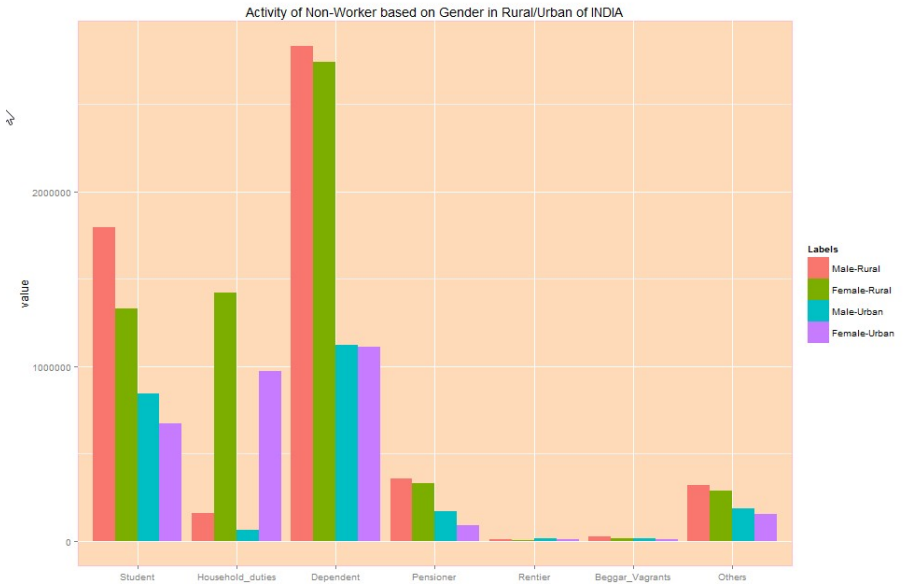
3.6 Test Results of analysis of non-workers based on Gender

Below is the screen shot of the execution of the test run. We have chose choice as character b and area_name as india

R

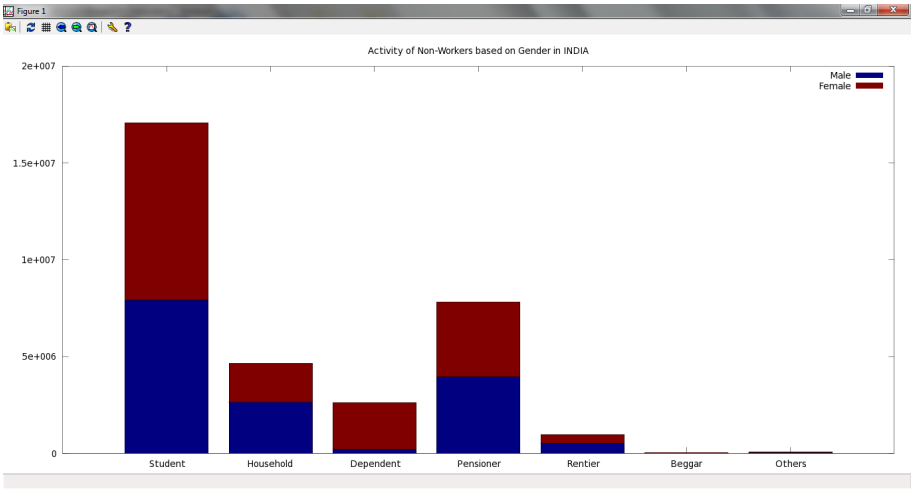


So it is clear the there are more number of men non-workers in India than women in all the categories.



So it is clear the there are more number of non-workers in all the categories in rural India.

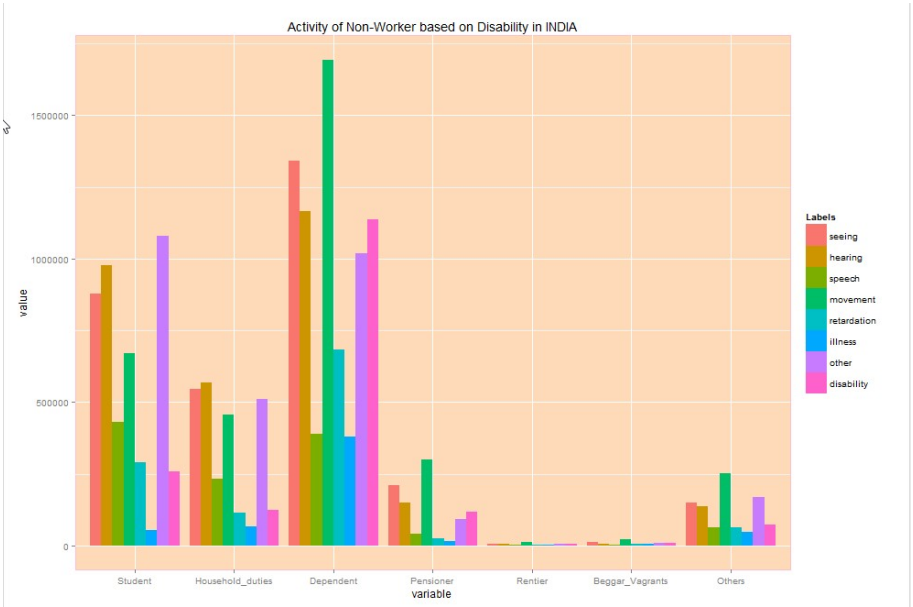
Octave



3.7 Test Results for Analysis based on Disabilities

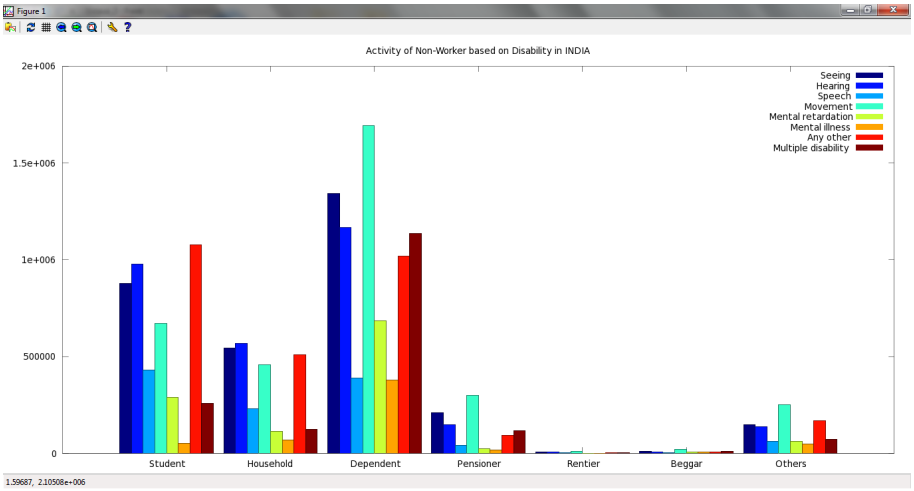
Below is the screen shot of the execution of the test run. We have chose choice as character c and area_name as india

R



It is clear in most the category type of the non-workers the major cause of disability is movement closely followed seeing and hearing.

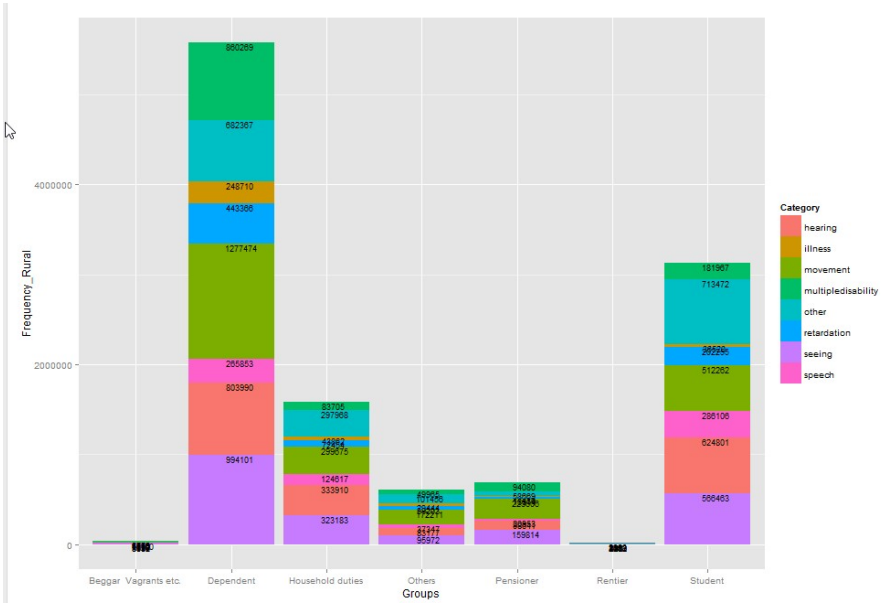
Octave



3.8 Test Results for Analysis based on Rural areas

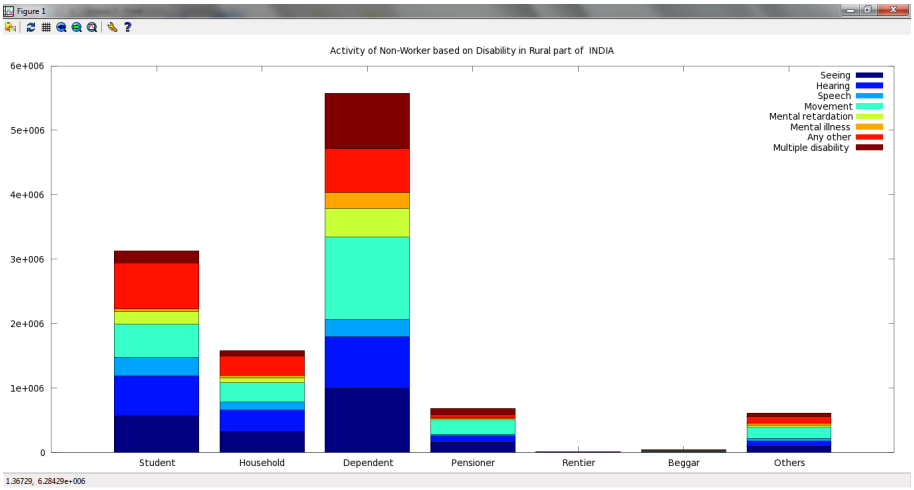
Below is the screen shot of the execution of the test run. We have chose choice as character d and area_name as india

R



So the major activity of non-workers in rural India is the dependent and the major factor for it due to the disability in seeing. Next major activity of non-workers in rural India is the dependent and the major factor for it due to the disability which is not known(other).

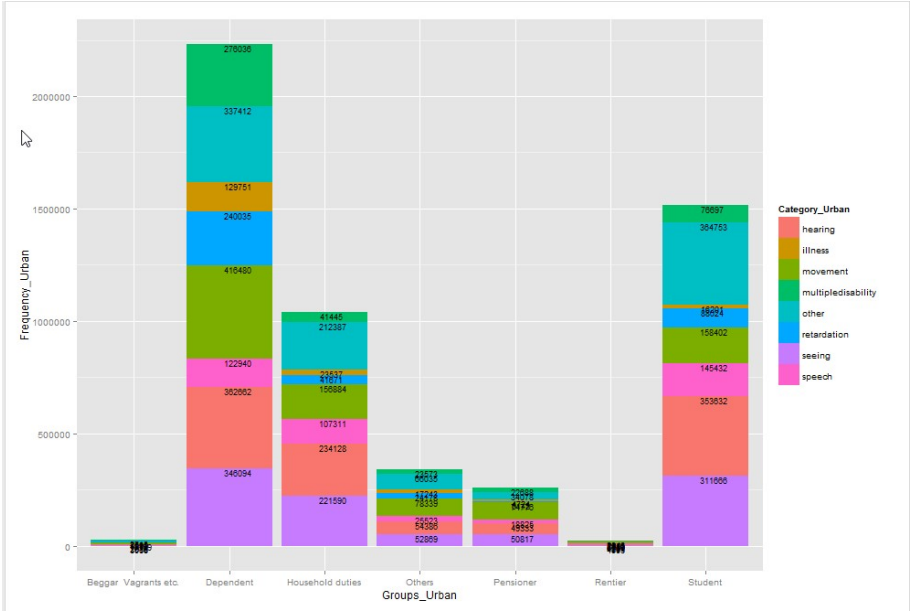
Octave



3.9 Test Results for Analysis based on Urban areas

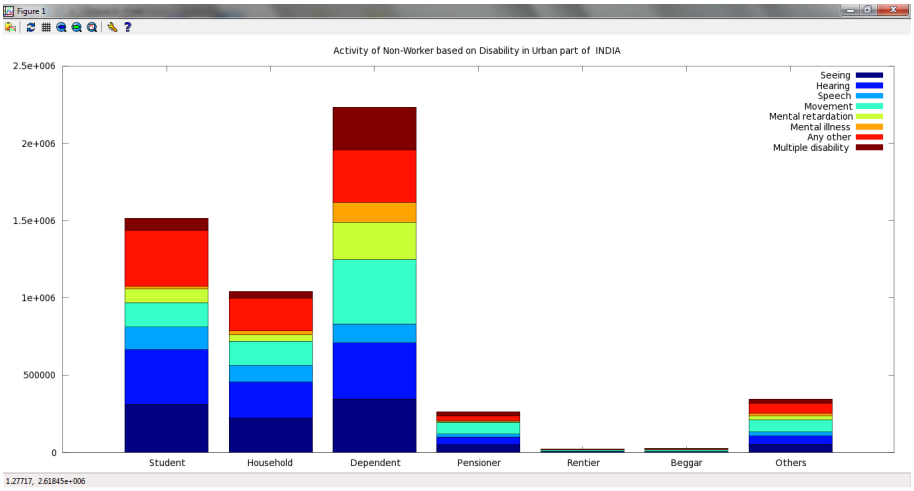
Below is the screen shot of the execution of the test run. We have chose choice as character e and area_name as india

R



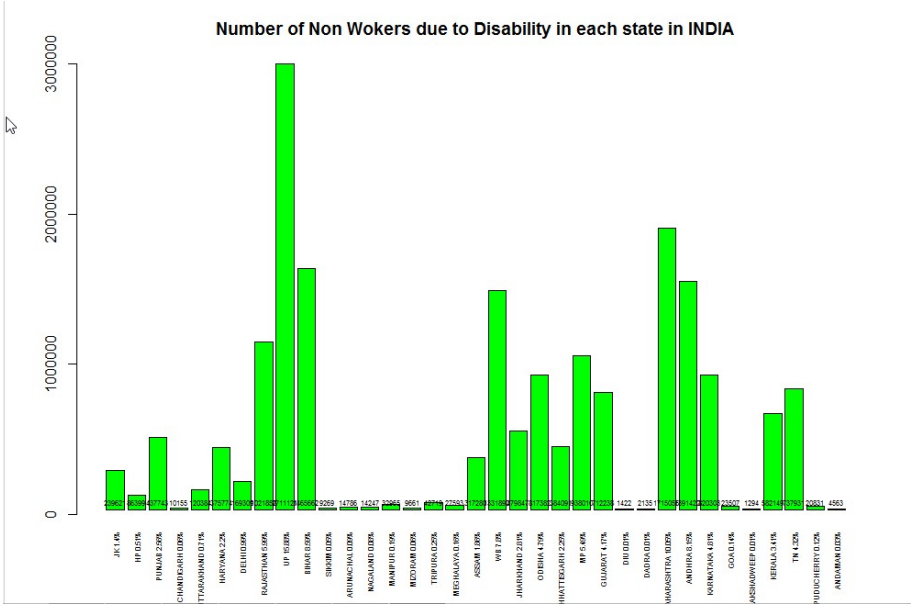
So the major activity of non-workers in urban India is the dependent and the major factor for it due to the disability in seeing. Next major activity of non-workers in rural India is the dependent and the major factor for it due to the disability which is not known(other).

Octave



3.10 Test Results of Case z

Below is the screen shot of the execution of the test run. We have chose choice as character z and area_name as india



So we can see the state of Uttar Pradesh is ranked 1 in terms of the disabled non-workers in India, this state has about 15.88 percent of the total disabled non-workers in India.Uttar Pradesh is followed by Maharashtra and Bihar.

4 Conclusion

In this project we have successfully analysed the difference between R and Matlab equivalent Octave. With the Disabled Non-Workers dataset we have shown several test results by plotting bar graphs and pie charts .

References

[1] Ramsay J.O, Hooker G, Graves S *Functional Data Analysis with R and Matlab* 2009, ISBN 9780387981857.

[2] David Hiebeler *MATLAB / R reference* June 24 2014. http://www.math.umaine.edu/ hiebeler/comp/matlabR.pdf

[3] Bitao Liu, Duncan Temple Lang *RMatlab Interface* Jan 15 2007 http://www.omegahat.org/RMatlab/outline.pdf

[4] Ecaterina Coman, Matthew W. Brewster, Sai K. Popuri, and Andrew M. Raim, and Matthias K. Gobbert *A Comparative*

Evaluation of Matlab, Octave, FreeMat, Scilab, R HPCF201215 http://profs.sci.univr.it/ caliari/pdf/octave.pdf