10/16/2016

Health Insurance and Demand for Medical Care

Data Analytics using R

Arpita Majumder,
Naresh Vemula,
Pallavi Singh,
Shresta Balerao,
Ziou Zhang

Contents

Executive Summary:

United States medical care expenditures account have grown about 4 percent per year and growing share of GDP and policy makers continue to search for mechanisms to rein in expenditure growth. In this environment, understanding the demand for medical care is critical. Unfortunately, estimating medical care demand is particularly challenging. One of the central problems is that the marginal price of medical care faced by consumers is often determined by consumers through their selection of a health insurance plan. In this report by estimating the overall risk of health care system, our team has decided to predict the annual medical expenditures excluding dental and outpatient mental paid by the sponsor or policy-holder to the health plan to purchase health coverage. For this estimation, we have performed data modelling such as lasso and ridge regression which helped us to penalized some of the parameters and determining the several factors that affect the medical expenditures. For instance, the least healthy individuals may be more likely to choose a plan with the most generous insurance coverage, leading to an overestimate of the effect on medical care demand. We also found that females are more likely to suffer from chronic diseases and to seek help for these conditions, females are spending more on medical care in comparison to males. Depending on health variable, it appears that people with poor health condition are paying more to the health providers. Apart from them, our team has decided to create a custom package to avoid the iterative use of functions to check the performance metrics for models and we have developed a user friendly shiny application by which user can calculate their medical expenditure by providing significant factors on screen such as age, annual income, gender, health etc. and would help them in maintaining their expenditure.

Data Overview:

The dataset ‘MedExp’ has been extracted from the package named ‘Ecdat’ which contains 15 variables and 5574 observations.

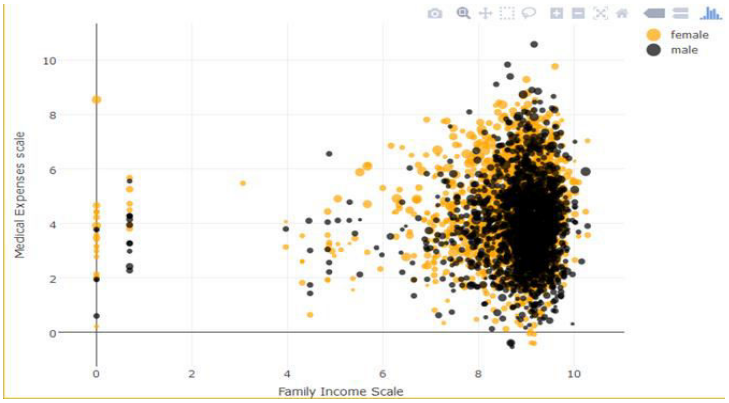
Variable Name	Data Dictionary	Values
med	annual medical expenditures in constant dollars excluding dental and outpatient mental	62.07547
lc	log(coinsrate+1) where coinsurance rate is 0 to 100	0
idp	individual deductible plan?	yes
lpi	log (annual participation incentive payment) or 0 if no payment	6.907755
fmde	log (max (medical deductible expenditure)) if IDP=1 and MDE>1 or 0 otherwise	0
physlim	physical limitation?	no
ndisease	number of chronic diseases	13.73189
health	self–rate health (excellent, good, fair, poor)	good
linc	log of annual family income (in \$)	9.528776
lfam	log of family size	1.386294
educdec	years of schooling of household head	12
age	exact age	43.87748
sex	sex (male, female)	male
child	age less than 18?	no
black	is household head black?	no

In order to understand the insurance marketplace, we have considered the rate and network files from Kaggle which contain data on health and dental plans offered to individuals and small businesses.

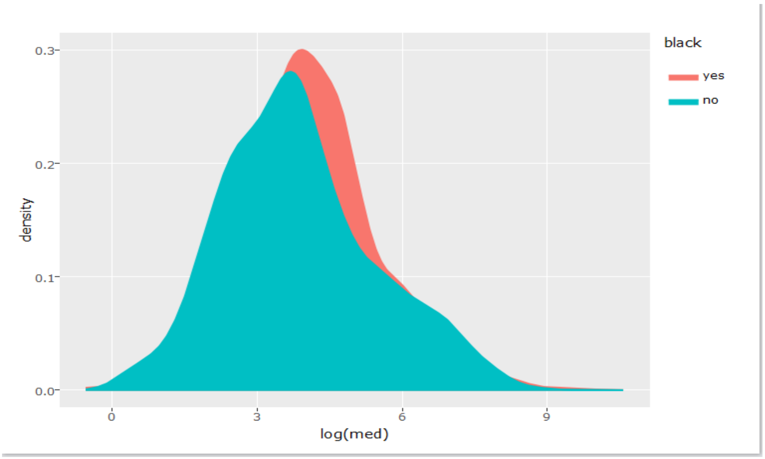
Data Exploration:

The dataset MedExp do not contain any missing values. After a thorough examination of the dataset, we could plot few interesting visualizations as shown below:

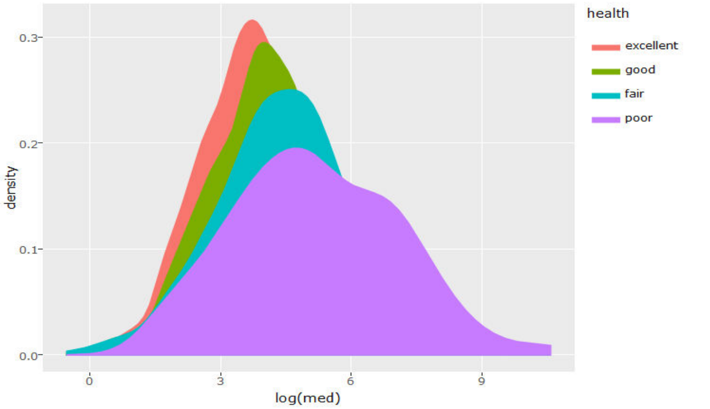
Packages used: **ggplot2 and plotly**



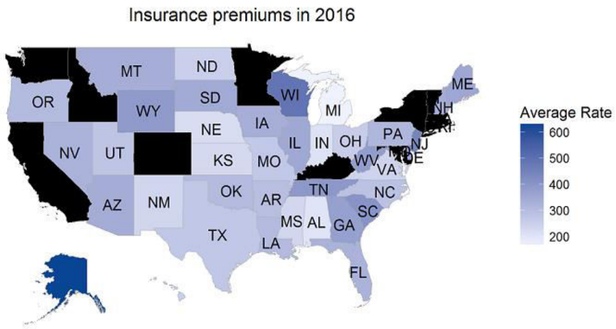
- We observe that the number of chronic diseases for a female are high as compared to that of males.
- On contrary males have high income than females but still females are spending more on medical expenses.



Depending on the race of the people, we observed that African-American people are paying more for their medical expenses as compared to others.



- One very obvious fact what we had seen was the variation of medical expenses with respect to the health condition.
- However, we observed that the amount of money spent for medical expenses is more for good health condition than fair health condition people. People having excellent health condition pay the least for medical expenses compared to others.



This map depicts the variation of the Insurance premiums in the year 2016 across different states in the US. We had seen that Wisconsin has highest insurance premium and the least is offered in Michigan state.

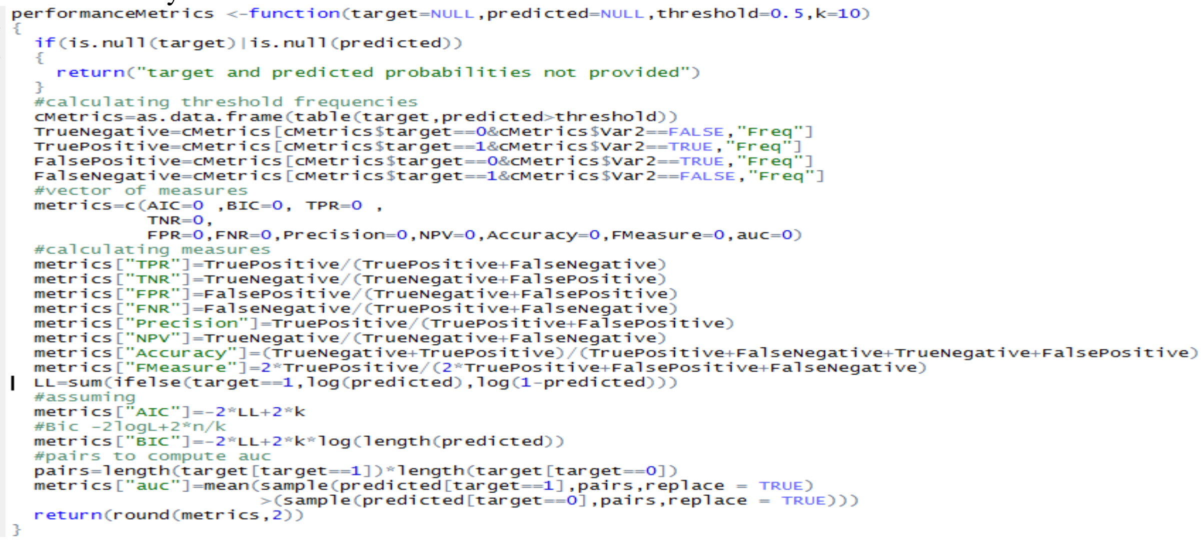
Modelling

Custom Package:

Our team has performed modelling to find the risk of incurring medical expenses among individuals. Simply, building a predictive model is not our motive. But, creating and selecting a model which gives high accuracy on out of sample data is our real objective. Hence, it is crucial to check accuracy of the model prior to computing predicted values. There are different kinds of metrics to evaluate our models. The choice of metric completely depends on the type of model and the implementation plan of the mode, so our team has developed a suite of functions to compute the accuracy metrics of the model. Because of the nature of iterative development, we have often reuse the functions many times, so we decided to create our own package. A Package that will bundles together code, data, documentation, and tests, and is easy to share with other. Functions and objects contained in a package and installed on a machine can be easily loaded using library.

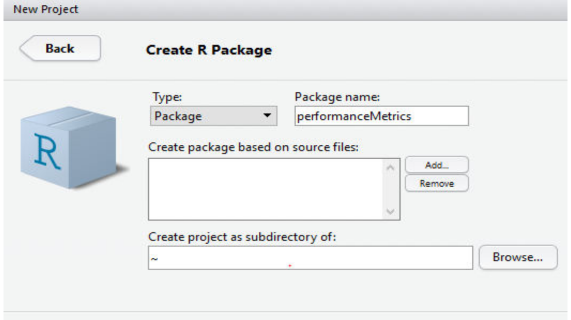
Basic procedure to create a package in R as follows –

- We have written the code for functions required for model evaluation and save it as “confusionMetrics” in .R file in our directory.

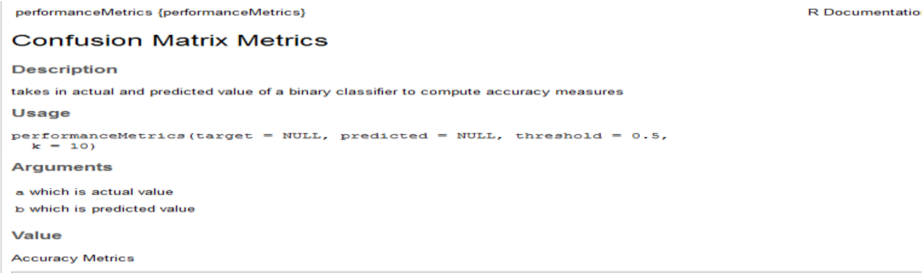


```
performanceMetrics <-function(target=NULL,predicted=NULL,threshold=0.5,k=10)
{
  if(is.null(target)|is.null(predicted))
  {
    return("target and predicted probabilities not provided")
  }
  #calculating threshold frequencies
  cMetrics=as.data.frame(table(target,predicted>threshold))
  TrueNegative=cMetrics[cMetrics$target==0&cMetrics$Var2==FALSE,"Freq"]
  TruePositive=cMetrics[cMetrics$target==1&cMetrics$Var2==TRUE,"Freq"]
  FalsePositive=cMetrics[cMetrics$target==0&cMetrics$Var2==TRUE,"Freq"]
  FalseNegative=cMetrics[cMetrics$target==1&cMetrics$Var2==FALSE,"Freq"]
  #vector of measures
  metrics=c(AIC=0 ,BIC=0, TPR=0 ,
            TNR=0,
            FPR=0,FNR=0,Precision=0,NPV=0,Accuracy=0,FMeasure=0,auc=0)
  #calculating measures
  metrics["TPR"]=TruePositive/(TruePositive+FalseNegative)
  metrics["TNR"]=TrueNegative/(TrueNegative+FalsePositive)
  metrics["FPR"]=FalsePositive/(TrueNegative+FalsePositive)
  metrics["FNR"]=FalseNegative/(TruePositive+FalseNegative)
  metrics["Precision"]=TruePositive/(TruePositive+FalsePositive)
  metrics["NPV"]=TrueNegative/(TrueNegative+FalseNegative)
  metrics["Accuracy"]=(TrueNegative+TruePositive)/(TruePositive+FalseNegative+TrueNegative+FalsePositive)
  metrics["FMeasure"]=2*TruePositive/(2*TruePositive+FalsePositive+FalseNegative)
  LL=sum(ifelse(target==1,log(predicted),log(1-predicted)))
  #assuming
  metrics["AIC"]=-2*LL+2*k
  #Bic -2logL+2*n/k
  metrics["BIC"]=-2*LL+2*k*log(length(predicted))
  #pairs to compute auc
  pairs=length(target[target==1])*length(target[target==0])
  metrics["auc"]=mean(sample(predicted[target==1],pairs,replace = TRUE)
                    >(sample(predicted[target==0],pairs,replace = TRUE)))
  return(round(metrics,2))
}
```

- The packages you will need to create a package are “devtools” and “roxygen2”, so we have installed these 2 packages. We have created a new project from file menu -New project -New Directory - R Package and have named it as “performanceMetrics”.



- File directory will contain a performance Metrics folder contains the code of our functions and ‘man’ folder will contain the help files for each function in the package. We have created the documentation for our function which will explain the functionality.



performanceMetrics {performanceMetrics} R Documentatio

Confusion Matrix Metrics

Description

takes in actual and predicted value of a binary classifier to compute accuracy measures

Usage

```
performanceMetrics(target = NULL, predicted = NULL, threshold = 0.5,
  k = 10)
```

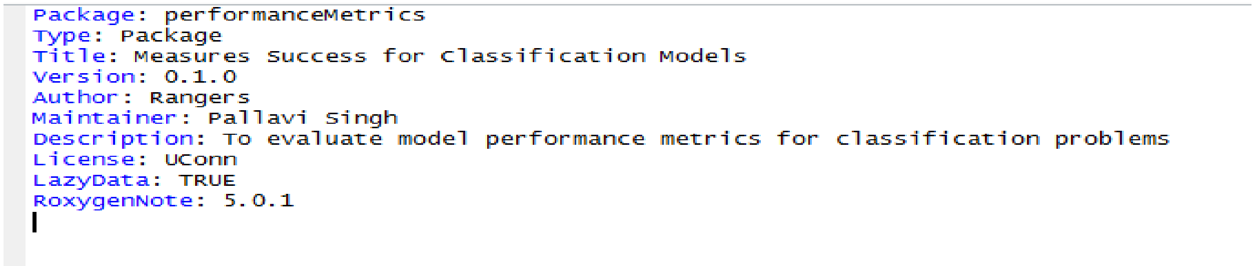
Arguments

a which is actual value
b which is predicted value

Value

Accuracy Metrics

- Create a documentation for what does package does -



```
Package: performanceMetrics
Type: Package
Title: Measures Success for Classification Models
Version: 0.1.0
Author: Rangers
Maintainer: Pallavi Singh
Description: To evaluate model performance metrics for classification problems
License: UConn
LazyData: TRUE
RoxygenNote: 5.0.1
```

Function Description- For classification model evaluation metric discussion, we have calculated below performance vectors while assuming a threshold of 0.5.

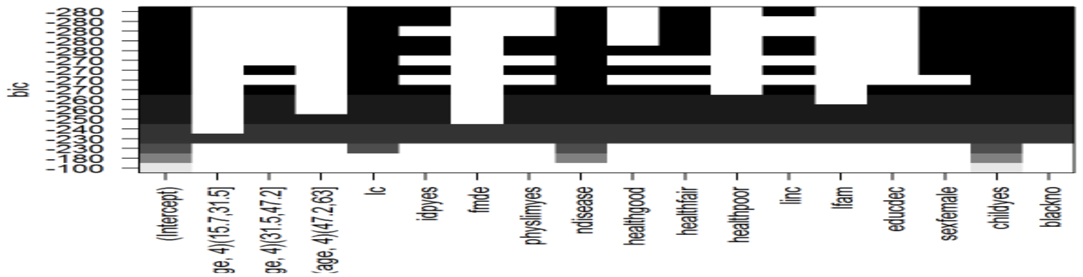
1. **Confusion Matrix** - A confusion matrix is an N X N matrix, where N is the number of classes being predicted. Below variables are required to create the confusion matrix.

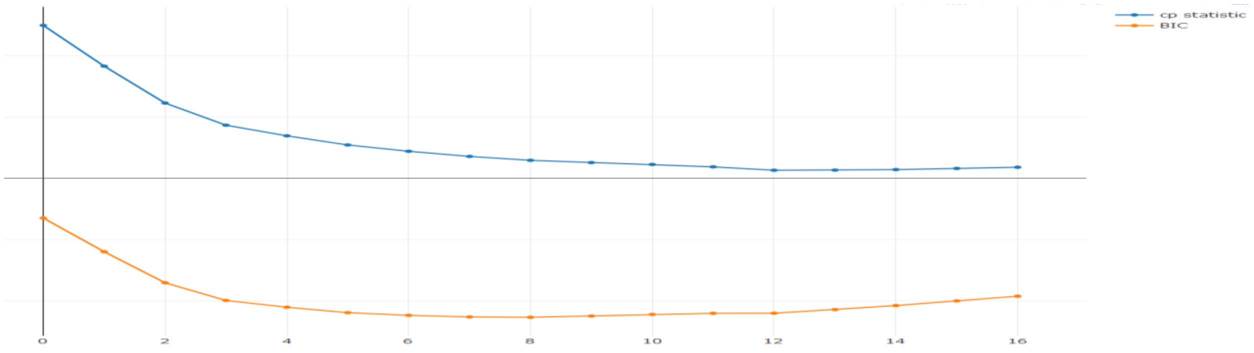
- **Accuracy**: the proportion of the total number of predictions that were correct.
- **Positive Predictive Value or Precision**: the proportion of positive cases that were correctly identified.
- **Negative Predictive Value**: the proportion of negative cases that were correctly identified.
- **Sensitivity or Recall**: the proportion of actual positive cases which are correctly identified.
- **Specificity**: the proportion of actual negative cases which are correctly identified.
- **FMeasure** – It is a measure of test accuracy, it considers both the precision and recall. F score reaches its value at 1 and worst at 0.
- **AIC & BIC** – These two values used to compare the various models for the same dataset to determine the best-fiiting model.The model having the smallest value is usually preferred.
- **AUC** – The AUC is a common evaluation metric for binary classification problem. if the classifier is very good, the true positive rate will increase quickly and the area under the curve will be close to 1.

Linear and Polynomial Models:

Linear model selection: We have built models using linear regression taking med as our dependent variable. But dataset is small so we have decided not to lose data by splitting further into validation dataset. Results from linear and polynomial model were incorporated below. To reduce the bias and variance in our predictions, we moved on to use 'regsubsets' function in 'leaps' package to do linear model selection methods using stepwise, forward and backwards selection models. By comparing cp statistic and BIC we identified the best model with 8 variables.

Variable selection using the BIC measure for feature selection





Further to see if we can improve models efficiency and interpretability we have used ridge and lasso regularization to reduce variance in the model.

Regularization: GLMNET package in r is used to perform generalized linear model with penalized maximum likelihood estimate. The regularization is performed by using algorithms such as Ridge, Lasso, and elastic net penalties. GLMNET can be used to perform linear regression, Binomial regression, multinomial regression and COX proportional hazards regression for survival analysis.

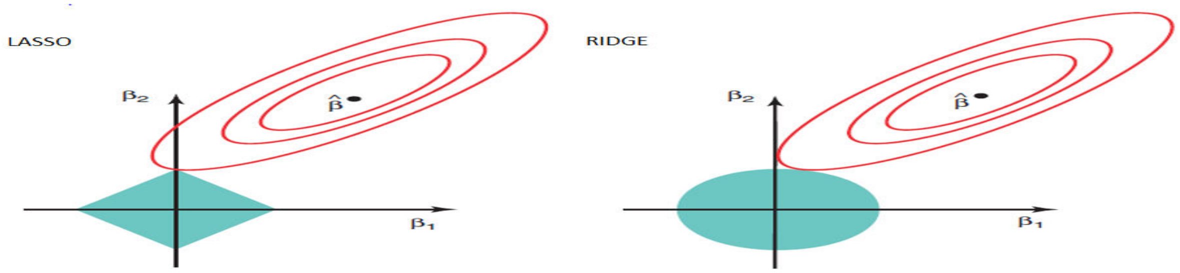
Ridge Penalty (L2 Norm Penalty): Ridge penalty reduces the bias by making the coefficients towards zero but never zero.

L2 Norm= $(\frac{1-\alpha}{2})\sum\beta^2 \leq t$

Lasso Penalty: Lasso penalty because of it linear additive form it reduces few of the coefficients to zero.

L1 Norm= $\alpha\sum\beta \leq t$

Where β are the model coefficients



Left figure shows the lasso penalty (L1 Norm), we can see that it touches the axis exactly which makes few of the coefficients as 0's. whereas it is different with respect to the ridge penalty (L2 Norm)

Model Building: For building a binomial logistic regression model in GLMNET we used GLMNET function, this function has set of parameters which are discussed briefly below.

alpha=0 triggers ridge regression using L2 Norm by making L1 norm to 0

alpha=1 triggers Lasso regression using L1 Norm by making L2 norm to 0.

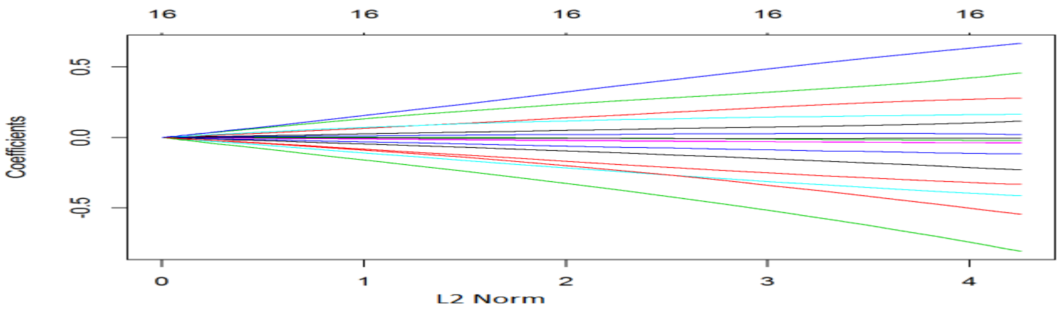
(1-α)/2||β||_2^2+α||β||_1

we will use lambda parameter as tuning parameter with a grid of 100 values from 10 ^ (10, -2)

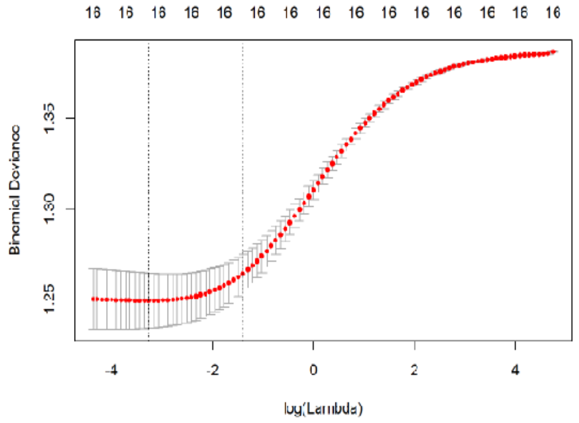
standardize=TRUE is set automatically, if we want to override it or else set it to false

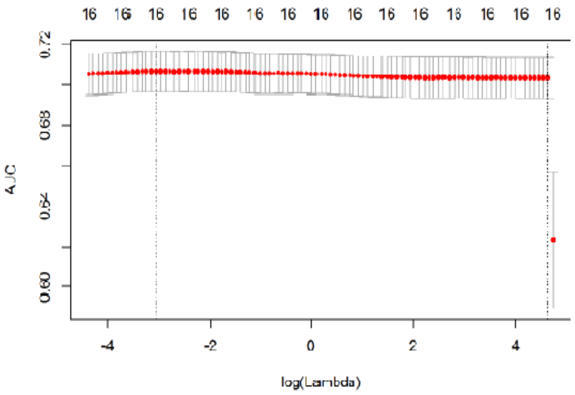
family= binomial or poisson or multinomial or cox

Ridge regression: We have modeled the same regression problem as a classification problem by categorizing the medical expenses into two bins, which are low and high. We computed a model matrix of independent variables and a dependent variable to send as inputs into GLMNET function. Using ridge, we can observe that the coefficients are close to zero when we have a small threshold and a high lambda value. This way by reducing the variables towards zero we have reduced the variance in the mode. But we can see 16 in the below picture indicating that we are considering all the variables in the model.

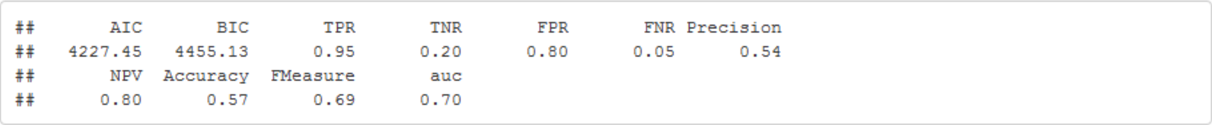


We have computed cross validation using error measure using cv. glmnet function from glmnet package. We have a parameter called type measure to perform cross validation with respect to the selected measure. After doing the cross validation we have selected the best lambda which is 0.01364037. For this lambda value, we have computed the performance metrics using the package we have developed.



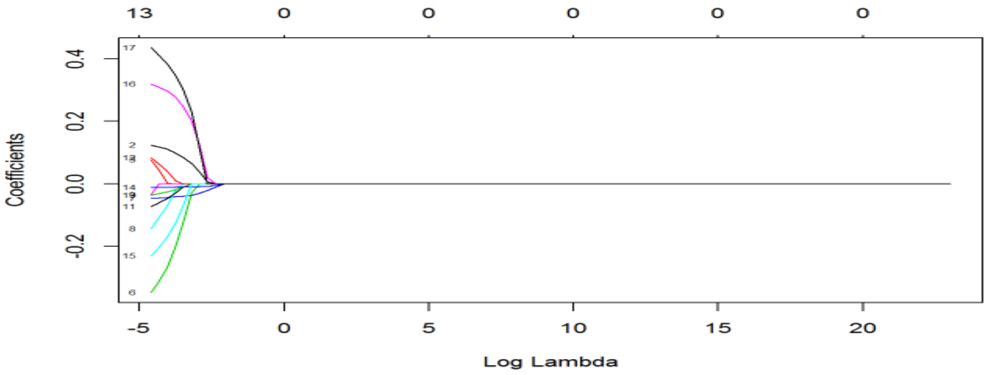


Performance metrics of the best Ridge Model

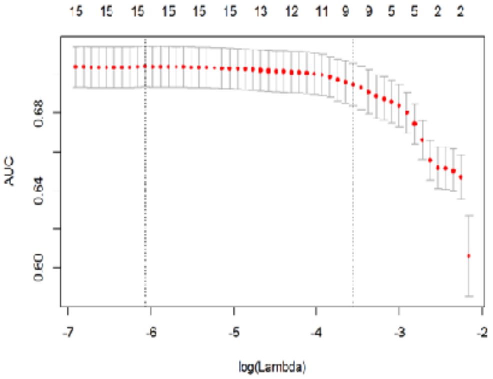


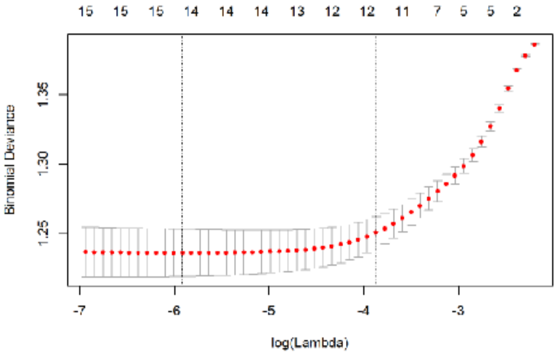
```
##         AIC       BIC        TPR        TNR        FPR        FNR Precision
##     4227.45   4455.13       0.95       0.20       0.80       0.05      0.54
##         NPV  Accuracy   FMeasure        auc
##        0.80      0.57       0.69       0.70
```

Lasso regression: By using lasso we have identified the key parameters which are affecting the medical expenses, we can observe from the below picture that all the coefficients are regressing to zeros with the increase in lambda value.

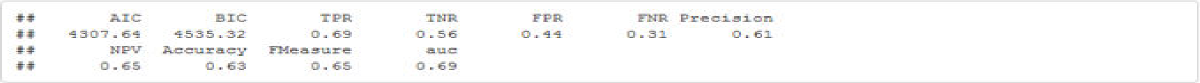


After doing the cross validation we have selected the best lambda for lasso is 0.01332677. However, we have not seen any big differences between lasso and ridge with this data set in terms of accuracy. But we have made few of the variables to z ero which increased our interpretability of the solution.

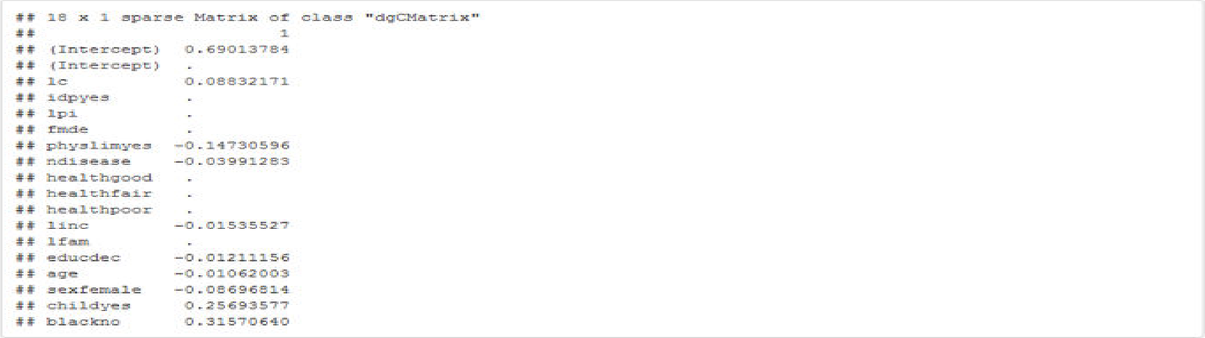




Using regularization, we observed that few of the variables are regressing to zeros and found significant factors such as number of disease, age, gender, disabilities, income and health conditions affect the medical expenses positively



```
##        AIC        BIC        TPR        TNR        FPR        FNR Precision
##    4307.64    4535.32       0.69       0.56       0.44       0.31      0.61
##        NPV   Accuracy   FMeasure        auc
##       0.65       0.63       0.65       0.69
```

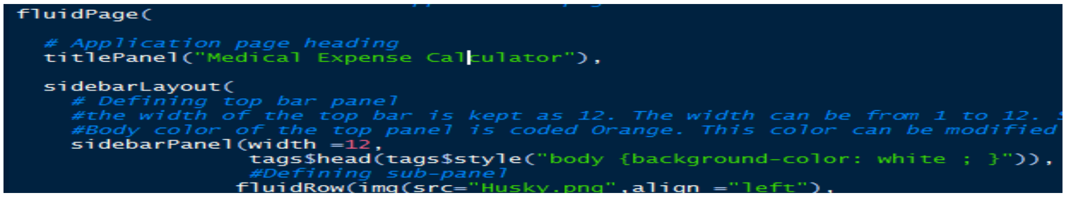


```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  0.69013784
## (Intercept)  .
## lc           0.08832171
## idpyes       .
## lpi          .
## fmde         .
## physlimyes  -0.14730596
## ndisease    -0.03991283
## healthgood   .
## healthfair   .
## healthpoor   .
## linc        -0.01535527
## lfam         .
## educdec     -0.01211156
## age         -0.01062003
## sexfemale   -0.08696814
## childyes     0.25693577
## blackno      0.31570640
```

Shiny Application: Medical Expense Calculator

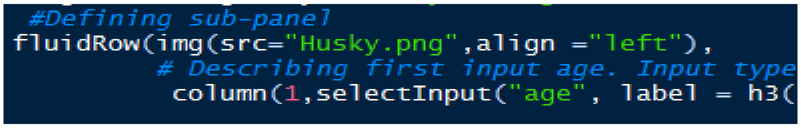
Shiny is an R package that makes it easy to build interactive web applications (apps) straight from R. Shiny app has three components:

User interface: shiny package needs to be included as a first command of the code.



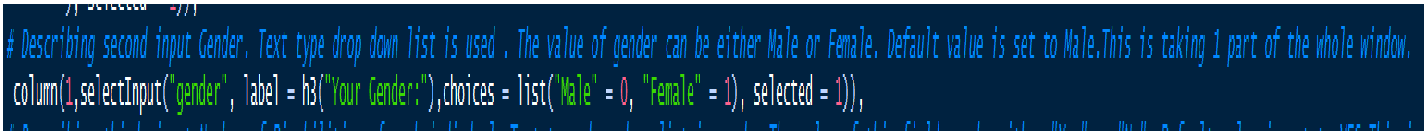
```
fluidPage(

  # Application page heading
  titlePanel("Medical Expense Calculator"),

  sidebarLayout(
    # Defining top bar panel
    #the width of the top bar is kept as 12. The width can be from 1 to 12.
    #Body color of the top panel is coded Orange. This color can be modified
    sidebarPanel(width =12,
                 tags$head(tags$style("body {background-color: white ; }")),
                 #Defining sub-panel
                fluidRow(img(src="Husky.png",align ="left"),
```

The first function is the fluidpage() holds the page layout. Each page will have sidebarpanel() and mainpanel() layout. Sidebarpanel() will contain input buttons like select button, dropdown button slider button, text/numeric input button etc.. Each button can be arranged as rows/column inside sidebar. The command for sidebar row will be as below:



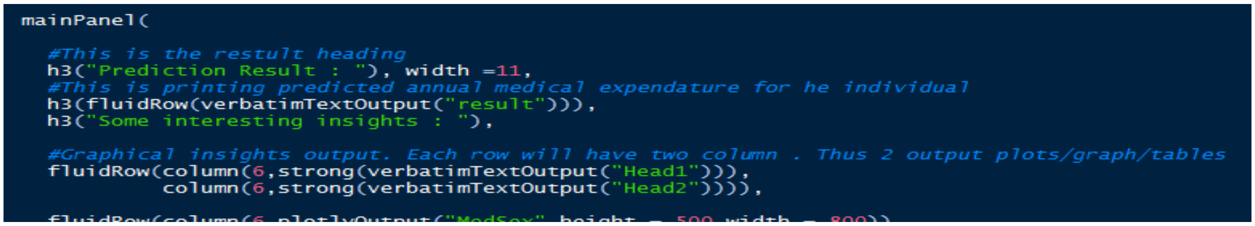
```
 #Defining sub-panel
fluidRow(img(src="Husky.png",align ="left"),
          # Describing first input age. Input type
           column(1,selectInput("age", label = h3(
```

The columns are mapped as:



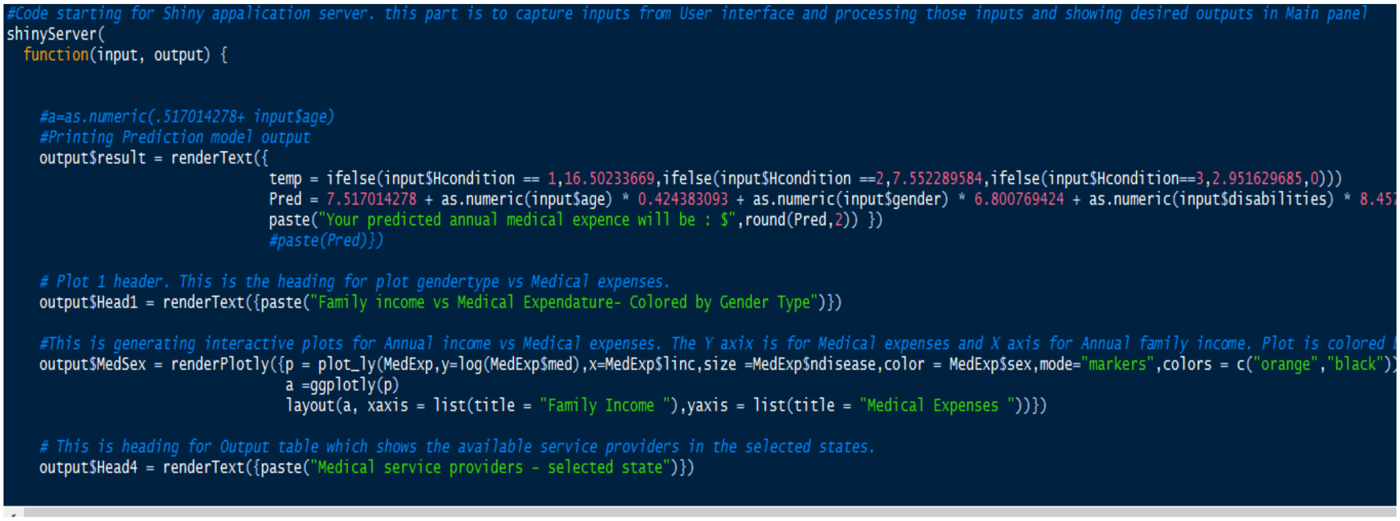
```
# Describing second input Gender. Text type drop down list is used . The value of gender can be either Male or Female. Default value is set to Male.This is taking 1 part of the whole window.
 column(1,selectInput("gender", label = h3("Your Gender:"),choices = list("Male" = 0, "Female" = 1), selected = 1)),
```

Each column will hold width value 1 to 12. Above screen shot shows that button for gender is holding 1unit place in the row. Similarly mainPanel() can also be divided into rows and expected outputs can be showed in the output panel.



```
mainPanel(

  #This is the restult heading
  h3("Prediction Result : "), width =11,
  #This is printing predicted annual medical expendature for he individual
  h3(fluidRow(verbatimTextOutput("result"))),
  h3("Some interesting insights : "),

  #Graphical insights output. Each row will have two column . Thus 2 output plots/graph/tables
  fluidRow(column(6,strong(verbatimTextOutput("Head1"))),
           column(6,strong(verbatimTextOutput("Head2")))),
```

<u>Server:</u> The server code will act as a function which will capture input values from UI and provide output to UI. Here is the sample code for server:



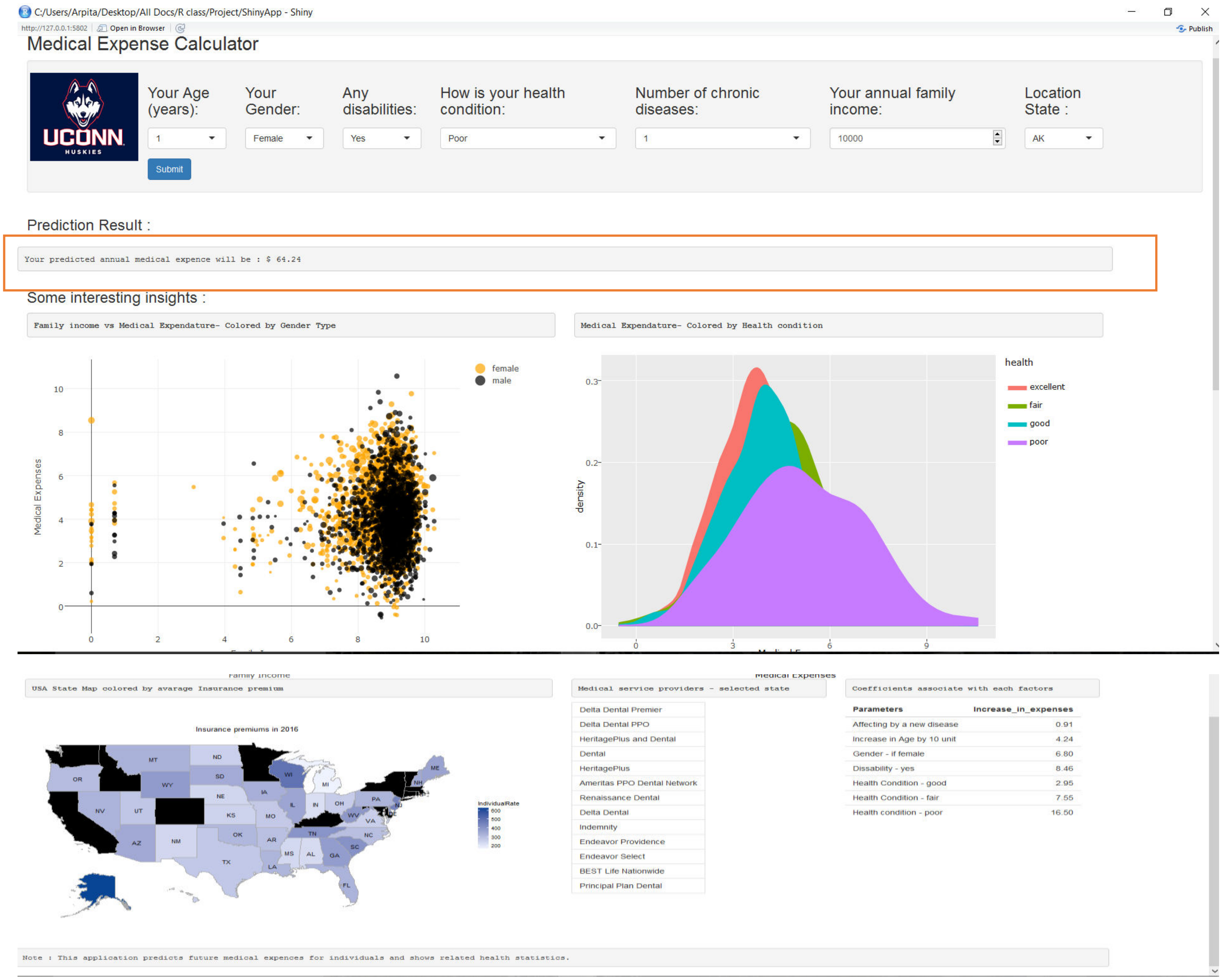
```
#Code starting for Shiny appalication server. this part is to capture inputs from User interface and processing those inputs and showing desired outputs in Main panel
shinyServer(
  function(input, output) {


    #a=as.numeric(.517014278+ input$age)
    #Printing Prediction model output
    output$result = renderText({
                                temp = ifelse(input$Hcondition == 1,16.50233669,ifelse(input$Hcondition ==2,7.552289584,ifelse(input$Hcondition==3,2.951629685,0)))
                                Pred = 7.517014278 + as.numeric(input$age) * 0.424383093 + as.numeric(input$gender) * 6.800769424 + as.numeric(input$disabilities) * 8.45
                                paste("Your predicted annual medical expence will be : $",round(Pred,2)) })
                                #paste(Pred)})

    # Plot 1 header. This is the heading for plot gendertype vs Medical expenses.
    output$Head1 = renderText({paste("Family income vs Medical Expendature- Colored by Gender Type")})

    #This is generating interactive plots for Annual income vs Medical expenses. The Y axix is for Medical expenses and X axis for Annual family income. Plot is colored
    output$MedSex = renderPlotly({p = plot_ly(MedExp,y=log(MedExp$med),x=MedExp$linc,size =MedExp$ndisease,color = MedExp$sex,mode="markers",colors = c("orange","black")
                                  a =ggplotly(p)
                                  layout(a, xaxis = list(title = "Family Income "),yaxis = list(title = "Medical Expenses "))})

    # This is heading for Output table which shows the available service providers in the selected states.
    output$Head4 = renderText({paste("Medical service providers - selected state")})
```

The output fields mention in the UI – main panel can be accessed in the server code. By rendertext/renderimage keyword all imputations can be made available in the output page.

<u>Running application:</u> a directory should be created with the name of the application("ShinyApp"). That directory will have UI code, Server code, www folder (containing pictures) and data folder (containing data file).

<u>Code to run application:</u> runApp("ShinyApp")

Here are the screen shot of the Medical Expense Calculator application made for this project:



Medical Expense Calculator

Your Age (years): 1

Your Gender: Female

Any disabilities: Yes

How is your health condition: Poor

Number of chronic diseases: 1

Your annual family income: 10000

Location State : AK

Prediction Result :

```
Your predicted annual medical expence will be : $ 64.24
```

Some interesting insights :

```
Family income vs Medical Expendature- Colored by Gender Type
```

```
Medical Expendature- Colored by Health condition
```

```
USA State Map colored by avarage Insurance premium
```

```
Medical service providers - selected state
```

Delta Dental Premier
Delta Dental PPO
HeritagePlus and Dental
Dental
HeritagePlus
Ameritas PPO Dental Network
Renaissance Dental
Delta Dental
Indemnity
Endeavor Providence
Endeavor Select
BEST Life Nationwide
Principal Plan Dental

```
Coefficients associate with each factors
```

Parameters	Increase_in_expenses
Affecting by a new disease	0.91
Increase in Age by 10 unit	4.24
Gender - if female	6.80
Dissability - yes	8.46
Health Condition - good	2.95
Health Condition - fair	7.55
Health condition - poor	16.50

```
Note : This application predicts future medical expences for individuals and shows related health statistics.
```

Above application takes 7 input variables: Age, Gender, Health condition, physical limitation, annual family income and state code. All of the variables other than state code are used to predict the future medical expense for the individual patients. State code is captured in order to display available medical service providers present in that particular state.

Users: This application is built keeping individual patients in mind. In healthcare domain it is very difficult to capture potential customers to get the insurance. This application will predict future medical expenses and also show the possible healthcare option they can take in order to reduce their future expenses.

Conclusion:

- We have identified key risk factors causing high medical bills, our shiny application will predict the medical bills based on few risk parameters. This will help individuals to plan their health coverage expenses in advance.
- For instance, least healthy individuals will be more likely to choose a plan with the most generous health insurance coverage leading to overestimate of the effect on medical core demand.

References –

- https://vincentarelbundock.github.io/Rdatasets/doc/Ecdat/MedExp.html
- https://www.rstudio.com/products/shiny/shiny-user-showcase/
- http://ase.tufts.edu/economics/papers/partha_deb.pdf
- https://www.bea.gov/papers/pdf/healthdemand.pdf
- https://www.growingfamilybenefits.com/disability-more-expensive-women/
- https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics/
- http://shiny.rstudio.com/tutorial/
- http://stackoverflow.com/
- http://ase.tufts.edu/economics/papers/partha_deb.pdf