CS 567

Computational Statistics

Project 1 - User Manual

Life Insurance Using R

Authors:
Chao HUANG LIN
41716747
Hermann YEPDJIO
40917845
Lubna ALZAMIL
4144208

Instructor:
Dr.Donald DAVENDRA

February 10, 2019

Contents

1 Introduction

Our first project is an implementation of some formulas used by insurance companies to set prices for the products they sale. The implementation uses the R language. It reads the input data from external files (.csv and .txt) in order to allow the end user to manipulate them easily. The implementation produces two types of output: numerical and graphical to give more flexibility to the user on how to analyze the results.

2 Getting Started

2.1 Prerequisites

2.1.1 Technical Skills

Running our scripts does not require any programming skills. However, the following skills are necessary:

- basic knowledge on how to operate a computer
- advanced knowledge in statistics in order to interpret the results

2.1.2 Software

- R

  In order to run our R scripts, R must be preinstalled on the system. If R is not present on the system, please download and install it from the following link https://cran.r-project.org/. It is free and available for Windows, Mac and Linux platforms.

  NOTICE: R is a very unstable language, therefore bugs are to be expected when running R scripts. However, to reduce the chances to get into these issues, we recommend to select a CRAN Mirror from the University of California, Berkeley when downloading R or any other R packages.

- RStudio

  After installing R, download and install the latest version of RStudio from the following link https://www.rstudio.com/products/rstudio/download/. It is free and available for Windows, Mac and Linux platforms. RStudio is a user friendly graphical user interface (GUI) that makes writing and running R scripts easier.

2.2 Open the scripts in RStudio

The scripts come in a zip file named "R.zip." Unzip this file, then copy or cut the resulting "R" folder and paste it into the "Documents" folder.

Notice: It is important that the "R" folder is being placed into the "Documents" folder, otherwise R will not be able to find the input files when the scripts are executed.
However, if you wish to copy the "R" folder somewhere else than in "Documents," you will have to set the working directory inside the scripts as well. Instructions on how to do that are described in section 2.4 below.

To load the scripts in RStudio, two methods can be considered:

- From outside RStudio

  - open the 'R' folder

  - double click on any of the file with the ".R" extension to see its content in RStudio.
  There are 2 such files: "life_table.R" which contains the important formulas and "project1.R" where we call the formulas from the previous file, perform all the calculations and display the results.

- From inside RStudio

  - open RStudio from the Windows "start" menu or from the Desktop

  - select "File" on the upper left corner, then select "Open File" from the drop-down menu. Doing so will bring up a little windows.

  - from there, browse to the "Documents" Folder, then open the "R" folder, then select the ".R" file which you want to see the content and click on "open"

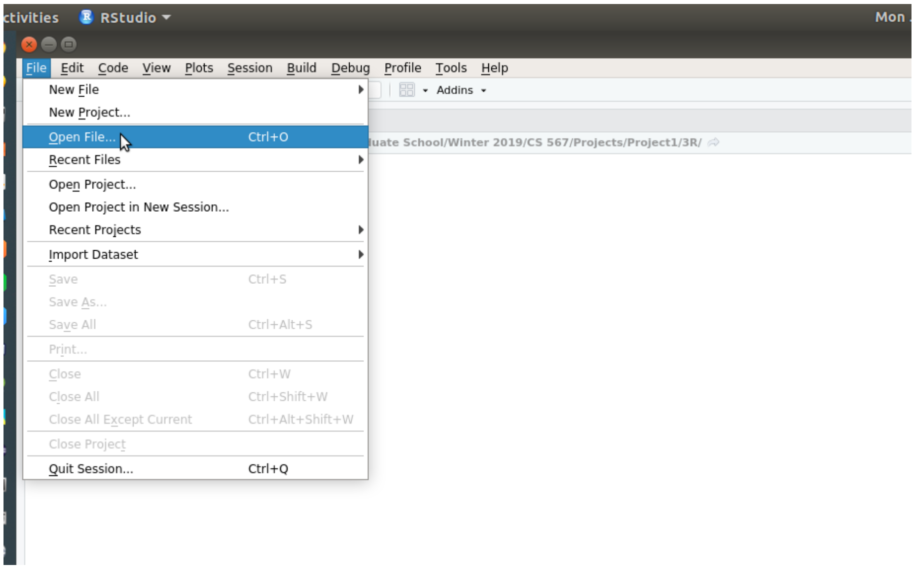
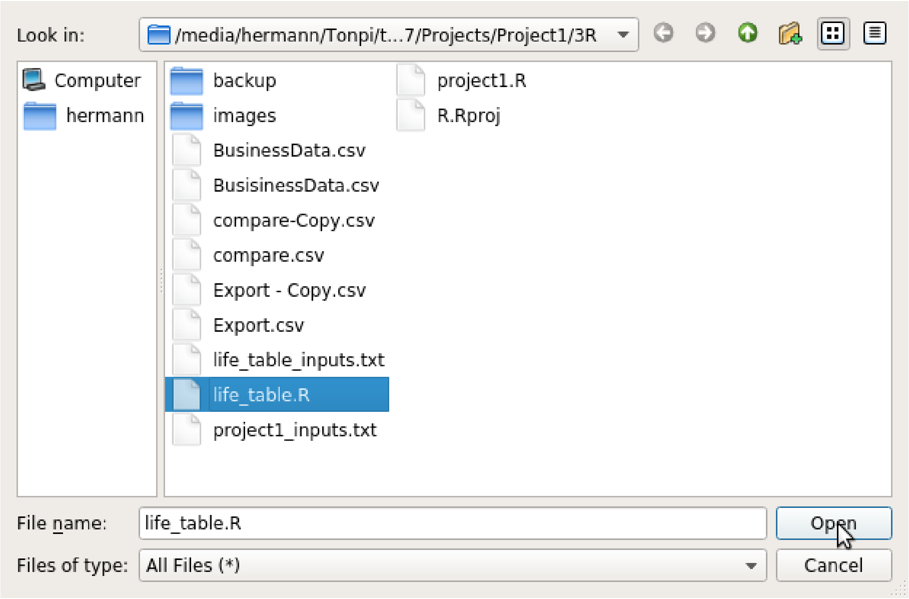


Figure 1: ”Open File” located in the ”File” drop down menu


Look in: /media/hermann/Tonpi/t...7/Projects/Project1/3R

File name: life_table.R

Files of type: All Files (*)

Figure 2: Browse to ”R” directory and open ”.R” files

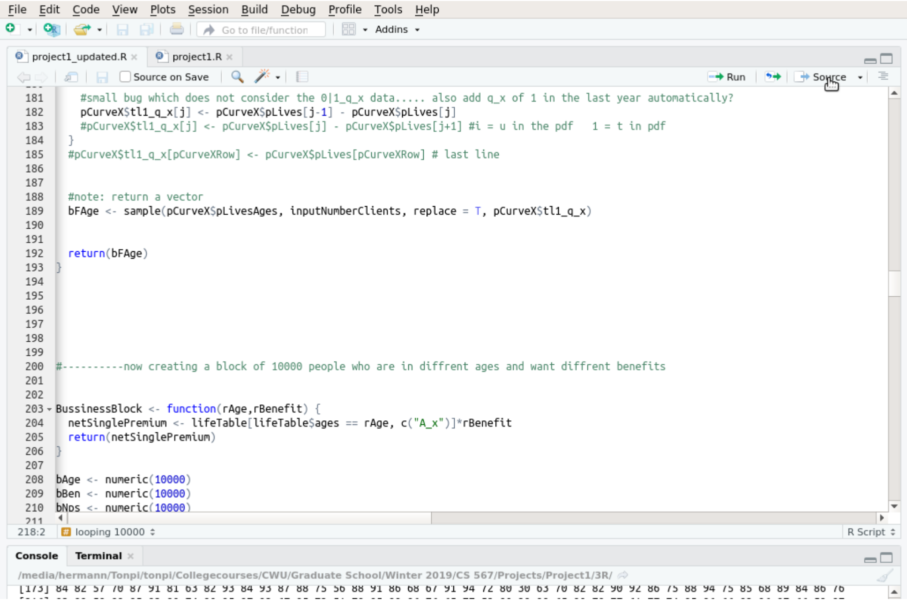
2.3 Run the scripts

As we have mentioned in the previous section, there are two R scripts in the ”R” folder. The one that needs to be run is named ”project1.R.” This script internally executes the second script called ”life_table.R” and produces all the expected results. So, to run the program two approaches can be considered:

Using the ”source” button

- first load the ”project1.R” file as described in the previous section
- then press the button ”source” as seen on figure 3 below.

Following this will run the whole code from the first line to the last line.



```
  #small bug which does not consider the 0|1_q_x data..... also add q_x of 1 in the last year automatically?
  pCurveX$tl1_q_x[j] <- pCurveX$pLives[j-1] - pCurveX$pLives[j]
  #pCurveX$tl1_q_x[j] <- pCurveX$pLives[j] - pCurveX$pLives[j+1] #i = u in the pdf   1 = t in pdf
 }
 #pCurveX$tl1_q_x[pCurveXRow] <- pCurveX$pLives[pCurveXRow] # last line


 #note: return a vector
 bFAge <- sample(pCurveX$pLivesAges, inputNumberClients, replace = T, pCurveX$tl1_q_x)


 return(bFAge)
}






#---------now creating a block of 10000 people who are in diffrent ages and want diffrent benefits


BussinessBlock <- function(rAge,rBenefit) {
 netSinglePremium <- lifeTable[lifeTable$ages == rAge, c("A_x")]*rBenefit
 return(netSinglePremium)
}

bAge <- numeric(10000)
bBen <- numeric(10000)
bNps <- numeric(10000)
```

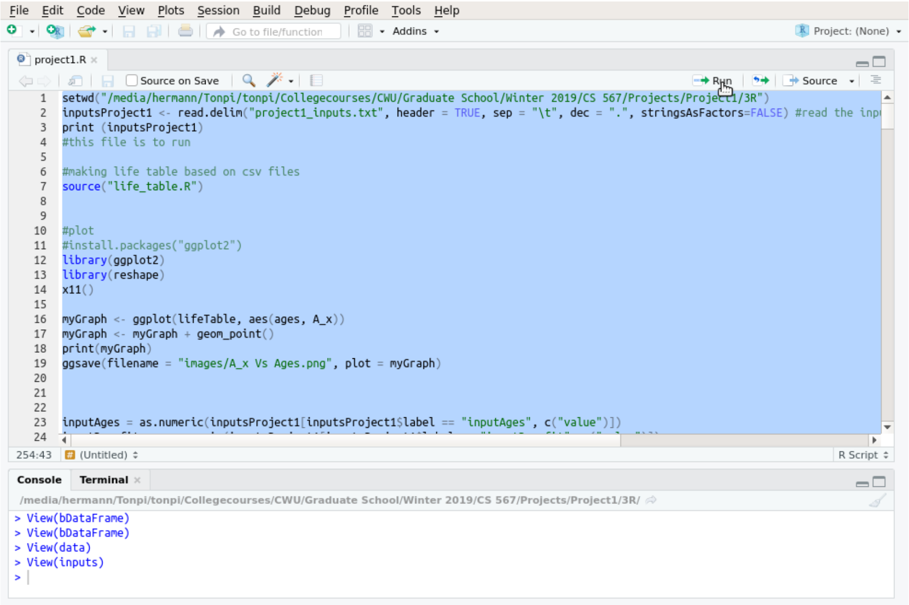
```
/media/hermann/Tonpi/tonpi/Collegecourses/CWU/Graduate School/Winter 2019/CS 567/Projects/Project1/3R/
[173] 84 82 57 70 87 91 81 63 82 93 84 93 87 88 75 56 88 91 86 68 67 91 94 72 80 30 63 70 82 82 90 92 86 75 88 94 75 85 68 89 84 86 76
```

Figure 3: running the script

Using the ”run” button

- first load the ”project1.R” file as described in the previous section

- press and hold the left button of the mouse, then drag the cursor to select the lines of code you wish to execute (or press ”ctrl + Alt” on the keyboard to select the entire script),
- then select ”run” as depicted on figure 4 below.



```r
setwd("/media/hermann/Tonpi/tonpi/Collegecourses/CWU/Graduate School/Winter 2019/CS 567/Projects/Project1/3R")
inputsProject1 <- read.delim("project1_inputs.txt", header = TRUE, sep = "\t", dec = ".", stringsAsFactors=FALSE) #read the inp
print (inputsProject1)
#this file is to run

#making life table based on csv files
source("life_table.R")


#plot
#install.packages("ggplot2")
library(ggplot2)
library(reshape)
x11()

myGraph <- ggplot(lifeTable, aes(ages, A_x))
myGraph <- myGraph + geom_point()
print(myGraph)
ggsave(filename = "images/A_x Vs Ages.png", plot = myGraph)



inputAges = as.numeric(inputsProject1[inputsProject1$label == "inputAges", c("value")])
```

```
/media/hermann/Tonpi/tonpi/Collegecourses/CWU/Graduate School/Winter 2019/CS 567/Projects/Project1/3R/
> View(bDataFrame)
> View(bDataFrame)
> View(data)
> View(inputs)
> 
```

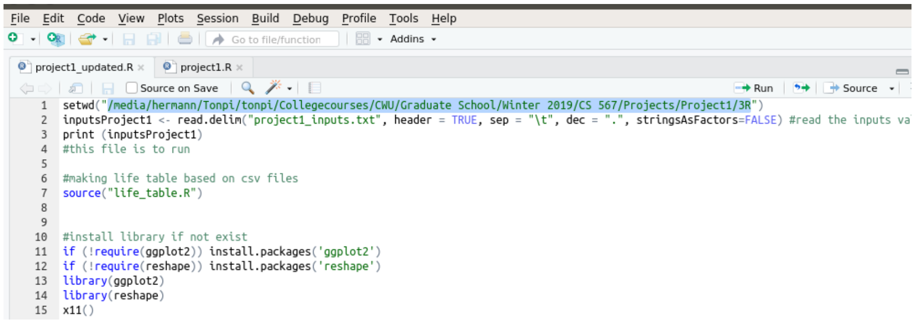
Figure 4: running the script

2.4 Changing the working directory

After loading both scripts into the R-Studio environment, look for the line that start with ”setwd” at the top of both scripts and replace the value inside the quotation mark by the path to the directory where you copied the scripts.(see figure below)

2.5 Additional Features

After running the script,



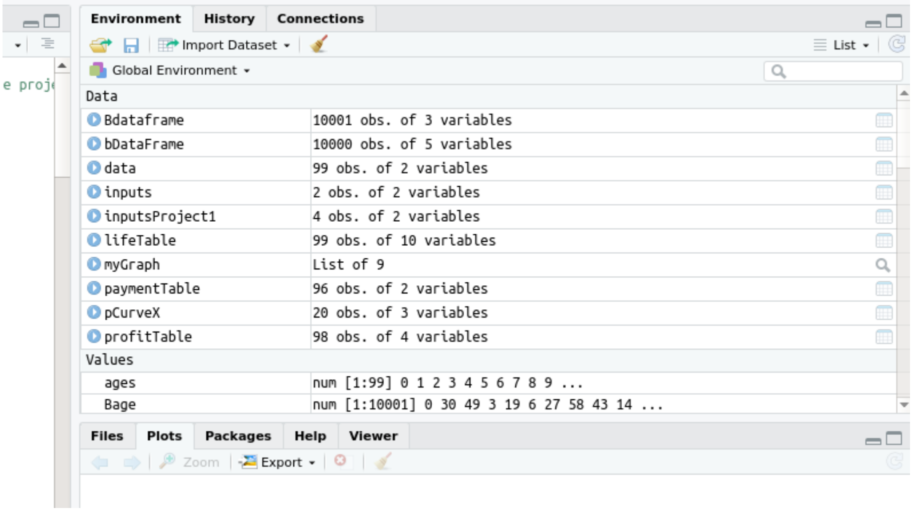
```r
setwd("/media/hermann/Tonpi/tonpi/Collegecourses/CWU/Graduate School/Winter 2019/CS 567/Projects/Project1/3R")
inputsProject1 <- read.delim("project1_inputs.txt", header = TRUE, sep = "\t", dec = ".", stringsAsFactors=FALSE) #read the inputs va
print (inputsProject1)
#this file is to run

#making life table based on csv files
source("life_table.R")


#install library if not exist
if (!require(ggplot2)) install.packages('ggplot2')
if (!require(reshape)) install.packages('reshape')
library(ggplot2)
library(reshape)
x11()
```

Figure 5: running the script

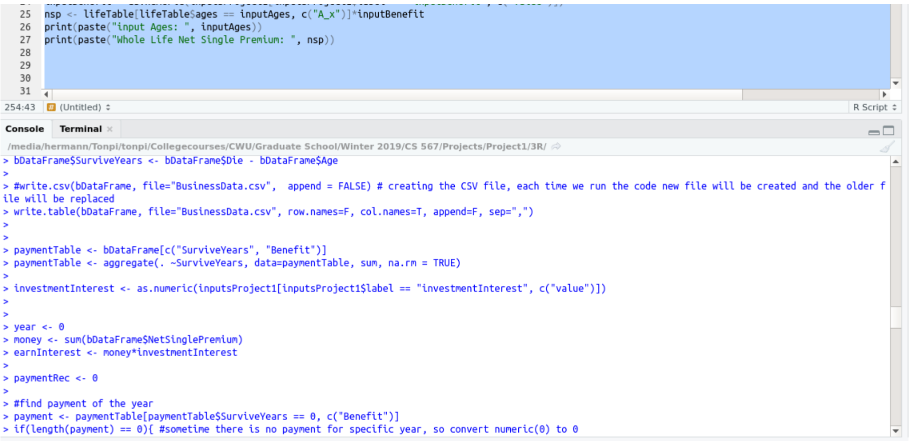
- you can get information about the data or input that was used by navigating through the little "Data" window on the upper right corner or the RStudio screen, and selecting the data which you need information about. (see figure 6.)



```
Data
Bdataframe       10001 obs. of 3 variables
bDataFrame       10000 obs. of 5 variables
data             99 obs. of 2 variables
inputs           2 obs. of 2 variables
inputsProject1   4 obs. of 2 variables
lifeTable        99 obs. of 10 variables
myGraph          List of 9
paymentTable     96 obs. of 2 variables
pCurveX          20 obs. of 3 variables
profitTable      98 obs. of 4 variables
Values
  ages           num [1:99] 0 1 2 3 4 5 6 7 8 9 ...
  Bage           num [1:10001] 0 30 49 3 19 6 27 58 43 14 ...
```

Figure 6: The Data window on the upper right corner of the screen

- you can see all the steps of the computations by checking the "Console" window located at the bottom left corner of the RStudio screen. (see figure 7.)


254:43
(Untitled)
R Script
Console
Terminal
/media/hermann/Tonpi/tonpi/Collegecourses/CWU/Graduate School/Winter 2019/CS 567/Projects/Project1/3R/

```
nsp <- lifeTable[lifeTable$ages == inputAges, c("A_x")]*inputBenefit
print(paste("input Ages: ", inputAges))
print(paste("Whole Life Net Single Premium: ", nsp))
```

```
> bDataFrame$SurviveYears <- bDataFrame$Die - bDataFrame$Age
>
> #write.csv(bDataFrame, file="BusinessData.csv",  append = FALSE) # creating the CSV file, each time we run the code new file will be created and the older f
ile will be replaced
> write.table(bDataFrame, file="BusinessData.csv", row.names=F, col.names=T, append=F, sep=",")
>
>
> paymentTable <- bDataFrame[c("SurviveYears", "Benefit")]
> paymentTable <- aggregate(. ~SurviveYears, data=paymentTable, sum, na.rm = TRUE)
>
> investmentInterest <- as.numeric(inputsProject1[inputsProject1$label == "investmentInterest", c("value")])
>
>
> year <- 0
> money <- sum(bDataFrame$NetSinglePremium)
> earnInterest <- money*investmentInterest
>
> paymentRec <- 0
>
> #find payment of the year
> payment <- paymentTable[paymentTable$SurviveYears == 0, c("Benefit")]
> if(length(payment) == 0){ #sometime there is no payment for specific year, so convert numeric(0) to 0
```

Figure 7: The Console window at the lower left corner of the screen

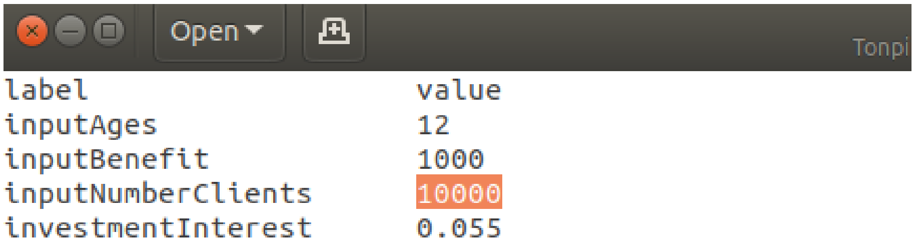
2.6 Accessing the Images

After running the scripts, different plots will be displayed on the screen automatically for visual analysis. However, copies of those images are also saved for later use in the ”Images” directory inside the ”R” folder. Those images are accessible at any time without the need to rerun the scripts.

3 Modify the inputs

All the variables and constants used in the scripts for all the calculations are input from text files (”.txt” extension) located in the R directory. There are two such files, ”life_table_inputs.txt” and ”project1_inputs.txt” which contain inputs for the ”life-table.R” and ”project1.R” scripts respectively.

To modify the inputs, open the text file containing the information you wish to change, change the value without changing the label and save the file. This change will be observed after the next execution of the scripts.



```
label                   value
inputAges               12
inputBenefit            1000
inputNumberClients      10000
investmentInterest      0.055
```

Figure 8: change input value

Notice: The highlighted value depicted on figure 8 is an example of a value that can be changed. The label in front of that value can not be changed, otherwise the script will not recognize the new label. If a value is changed, make sure to replace it with a new value of the same type (number, text etc...)