The User Guide to Pension Simulation Project

This pension simulation project is co-led by Dr. David S. T. Matkin and Dr. Gang Chen with the State University of New York at Albany. The project is funded by the Steven H. Sandell Grant Program, U.S. Social Security Administration via Boston College Center for Retirement Security. The purpose of the project is to simulate all kinds of normal cost and actuarial accrued liability based upon different possible actuarial assumptions with the chosen parameters. The final products of the projects include the R codes and seven CSV files, which can be accessed through the following link: https://tinyurl.com/m7kb3yf (will change to the URL of SUNY server once established).

The R codes are created to calculate all kinds of normal cost and actuarial accrued liability at the individual level based upon all possible actuarial assumptions within a defined parameter. There are sixteen parameters that can be adjusted in the code: 1. entry age; 2. retirement age; 3. current age; 4. minimum age limit of the model; 5. Maximum age limit of the model; 6. Inflation Rate; 7. past salary growth rate; 8. future salary growth rate; 9. Discount Rate; 10. Benefit Factor; 11. COLA (Cost of living adjustment); 12.AFC (average final compensation);13. Vesting Period; 14. Cost Calculation Methods; 15. Mortality Table Choices; 16. Starting Salary.

The created R codes include three major parts:

1. *Run_Simulation.R*, which is the main file to create data frame and is currently set to the default value of actuarial assumptions for creating the final data set of this project. The values of sixteen actuarial assumptions within line 50 to 100 can be adjusted to different requirements;

2. *Functions.R*, which is the file includes all related calculation[1] that *Run_Simulation.R* calls from*;*

3. *Generatecsv.R*, which is the code to create the final seven comma separated values (CSV) files using the data frame created by *Run_Simulation.R* including six separate data set, based upon the combination of two calculation choices [entry age normal (EAN) and projected unit calculation(PUC)] and three mortality table choices (mortality table 2000, 2010 and 2014), and a combined data set which consists of the six data set.

[1] Winklevoss, H. E. (1993). *Pension mathematics with numerical illustrations*. University of Pennsylvania Press.

The created seven CSV files:

1. The six CSV files are named by the calculation choices and the mortality table choices: *EANrp2000full, EANrp2010full, EANrp2014full, PUCrp2000full, PUCrp2010full, PUCrp2014full,* each of the six CSV files contain 967,861 observations each with the size of 85MB to 86MB;

2. The combined data set is the combined version of the six separate CSV files with a total of 5,807,202 individual observations and the size of the file is 514 MB.

To fully reproduce the result, all files in the folder named "R_Code" are required to be downloaded and unzipped into one newly created folder in addition to the most updated version of R [2]and RStudio [3](preferably coupling R studio with R version 3.3.3. released on Monday 2017-03-06).

In the folder "R_Code", there are three R codes files, one text file and another folder "Data" which includes two R data sets, *Mortality.RData and Termination.RData*.

The R studio is the preferred operating environment and the R codes can be directly used by R studio. The following steps are written for running the simulation in the RStudio environment. The downloaded codes can also be applied in the original R environment through coping all the contents from *R_Code.txt* and pasting to R console after adjusting the default saving route. The step of adjusting saving route can be done by substituting "XXX" part in the first line with the downloaded folder location and then type "enter."

The 12 Steps to Replicate or Create Your Own Pension Simulation Data Set:

1. Download and install R from website: https://www.r-project.org/
2. Download and install RStudio from website: https://www.rstudio.com/products/rstudio/
3. Install the following packages in RStudio or R first through typing "install.packages("packagename") in the console. The "packagename" should be replaced by the exact spellings of the following package names and then type "enter" to initiate the installing process;
   Or, click the button "Package" in one of the top sections within four panels and then click the button "Install. You need to type or copy the exact spellings to the blanks underneath "Packages." The installing process can be one by one or in one step through separating the package names with space or comma. More details instruction with examples about the installation process can be found in the following: https://www.r-bloggers.com/installing-r-packages/

[2] R can be downloaded at https://www.r-project.org/
[3] R studio can be downloaded at https://www.rstudio.com/products/rstudio/

Eight R Packages are required to be installed:

a. BayesBridge – a package to create different kinds of distributions of population;

b. dplyr – a package to manipulate data tasks within the model;

c. foreach – a package to enable for writing loops in R coding;

d. FinCal – a package to do financial calculation, time series analysis and computational finance etc.;

e. magrittr – a package to improve readability and maintainability of code;

f. rJava – a package to allow creation of objects and calling methods;

g. Scales – a package to enable number formatting and visualize data set used in the model.

h. XLConnect – a package to import mortality tables and termination rates as well as export the final csv files.

4. Download and unzip the whole folder from: https://tinyurl.com/m7kb3yf
5. Open the folder named “R_Code” within the newly downloaded file;
6. Open the *Run_Simulation.R* using RStduio by double clicking the file;
7. Select all the codes in the Source panel through right licking mouse within the source panel and click “Select All” in the poped up Common Tasks Bar or simply typing “Ctrl+A”;

   The code within line 50 to line 100 are the sixteen parameters are set to the default values and can be adjusted to your needs (see the further notes below);
8. Click “Run” button at the top right corner of the Source panel or simply type “Ctrl+R”;
9. The code will start to run (this process can take up to 30 hours depending on the chosen parameters and computing capacity of your computer) to generate the originally uploaded data set and it will generate a R data frame named *simulation_result.RData* in the same folder “R_Code”;
10. Once the step 9 is finished (refer to the console result and it will show the time when the computing is finished), open the Generatecsv.*R* in *the folder* using RStduio by double clicking the file;
11. Repeat the step 7 and 8 to run the whole coding within Generatecsv.*R*;
12. The codes will import the newly created R data frame *simulation_result.RData* and generate the final seven CSV files in the folder (see the further notes below). This step will take up to 2 hours depending on the computing capacity of your computer.

Note about creating your own pension simulation data set:

There are two steps involved to create your own pension simulation data set

A. In step 7, you can adjust the lines 50 to 100 within *Run_Simulation.R*

   There are following sixteen parameters that can be adjusted to your simulation needs and it can save many computing hours rather than the original 30 hours if you only generate some portion of the data set.

   1. Line 51: e_a<-seq(25,35,5), this defines the range of entry age to be 25 to 35 with 5 incremental increase within the chosen range, which is 25, 30, 35. You can adjust the range and the rate of incremental change;
   2. Line 54: r_a<-seq(55,65,5), this defines the range of retirement age to be 55 to 65 with 5 incremental increase within the chosen range, which is 55, 60, 65. You can adjust the range and the rate of incremental change;
   3. Line 57: ca<-seq(25,85,1), this defines the range of current age to be 25 to 85 with 1 incremental increase within the chosen range, which is 25, 26,…,84,85. You can adjust the range and the rate of incremental change;
   4. Line 60: min_age<-25, this defines the minimum age to be 25 in this model and you can adjust the value to your needs;
   5. Line 63: max_age<-85, this defines the maximum age to be 85 in this model and you can adjust the value to your needs;
   6. Line 66: inflation<-0.02, this defines annual inflation rate to be 0.02 or 2 percent in this model and you can adjust the value to your needs;
   7. Line 69: past_sgr<-seq(0.02,0.05,0.01)+inflation, this defines the range of past salary growth rate to be 0.02 to 0.05 with 0.01 incremental increase within the chosen range and the inflation rate 0.02, which makes the value to be 0.04, 0.05,…,0.07. You can adjust the range and the rate of incremental change;
   8. Line 72: future_sgr<-seq(0.02,0.05,0.01)+inflation, this defines the future expected salary growth rate to be 0.02 to 0.05 with 0.01 incremental increase within the chosen range and the inflation rate 0.02, which makes the value to be 0.04, 0.05,…,0.07. You can adjust the range and the rate of incremental change;
   9. Line 75: i_r<-seq(0.02,0.07,0.01)+inflation, this defines the future discount rate to be 0.02 to 0.07 with 0.01 incremental increase within the chosen range and the inflation rate 0.02, which makes the value to be 0.04, 0.05,…,0.09. You can adjust the range and the rate of incremental change;

10. Line 78: bf_p<-seq(0.01,0.04,0.01), this defines the benefit factor to be 0.01 to 0.04 with 0.01 incremental increase within the chosen range, which makes the value to be 0.01, 0.02, 0.03 and 0.04. You can adjust the range and the rate of incremental change;
11. Line 81: cola_r<-seq(0,0.04,0.01), this defines the COLA to be 0 to 0.04 with 0.01 incremental increase within the chosen range, which makes the value to be 0, 0.01, 0.02, 0.03 and 0.04. You can adjust the range and the rate of incremental change;
12. Line 84: afc_p<-5, this defines the AFC (average final compensation) to be 5 years. You can adjust the value to your needs;
13. Line 87: vesting<-5, this defines the Vesting Period to be 5 years. You can adjust the value to your needs;
14. Line 90: c_m<-c('EAN','PUC'), this defines the Cost Methods to be entry age normal (EAN) and projected unit calculation(PUC)]. You can keep both or only one of it;
15. Line 93: mort_num<-c(2,4,6), this defines the Mortality Table choices to be 2(RP2014), 4(RP2000) and 6(RP 2010) You can keep any one or two or all of them;
16. Line 96: sal<-1, this defines the starting salary to be $1 and You can adjust it your own numbers.

B. In step 12, you can adjust the lines from 5 to 19 based upon your previous choices cost methods and mortality table within *Generatecsv.R.* For example, If PUC is deleted, then line 8, 9, 10 should be deleted together with lines 15, 16, 17. Meanwhile, if mortality table 2010 is deleted, then line 6 and line 9 together with 13 and 16.