Student Loan Finance Model

Data Modelling and Analysis - Lecture 3

Dr Mercedes Torres Torres*

Background

Affording a place to study at university concerns all potential students. There may be fees to pay for the courses and there will certainly be a need for finance for cost-of-living expenses. Many students take up part-time employment whilst at university, sometimes working during term time as well as vacations. Support agencies offer help in a variety of ways with scholarships and bursaries.

Problem Statement

Study4Less, a Student Loans Company, can provide the maintenance aspect of student finance (e.g. food, transport, school materials, utilities, etc). The company loans students a given sum at the beginning of each of the three or four academic years covering the duration of the degree.

Repayment of the loan will start once the graduate is working and earning above a certain (cut-off) salary. Repayments are collected through National Insurance Contributions in the UK and the amount repaid by an individual depends on marginal income earned above the cut-off.

Study4Less loans include a 0.25% monthly interest over the compounded sum borrowed.

Problem description

Use mathematical modelling to solve these two problems:

1. Create a mathematical model to calculate, given a number of months, how much money a student will have to pay monthy.
2. Use your model to calculate how much a student who was in a 4-year programme and borrowed £2000 per annum from Study4Less will have to pay if they want to settle their debt in 10 years and interest .

Solution

In order to solve this problem, we refer back to the steps in model construction that we covered in Lecture 2:

1. Identify question or problem

*Adapted from Edwards and Hamson's *Guide to Mathematical Modelling*

2. System analysis
3. Formulate mathematical model
4. Solve mathematical model
5. Interpret solution

The idea is the following: we will look at the problem that we have to solve and the system that we have access to. After defining the variables that are available for us, we will come up with a generic model to solve the problem, and we will solve it with the case that has been given to us.

1. Identify question or problem

In this case, the problem has been given to us. With the information given, we can define Q.

Q: {*how can we calculate (or model) how much a student has to pay each month to pay off their debt to Study4Less in an allotted number of months?*}

2. System analysis

Let's define our general system, S. S can be quite general at this stage. After analysis S in detaill, we will come up with a list of variables and parameters that will form our *reduced system*, S_r.

S: {*The students' financial situation, amount of money borrowed, loan conditions, and time allotted to return the loan*}

From there, let us define the following variables and parameters for our S_r:

1. P: Initial (principal) amount borrowed (£)
2. r: interest charged per month (%)
3. n: Number of months count
4. N: Total number of months to complete the pay-off
5. M Repayment amount paid per month (£)

3. Formulate mathematical model

Some times, coming up with a complete mathematical formula for the model can be difficult. In those cases, you can start looking at the process that the model is supposed to, well, model, stage by stage. So, let's look at the repayment of the loan in stages.

At this stage, coming up with a formula to be able to calculate how much money the student must pay can be difficult.

However, what we can calculate relatively easily with the variables that we have is how much money is left to pay after each moth. At the end of each month, the amount left to pay is equivalent to the previous month's total plus interest minus one payment.

To make it even simpler, let's just look at the first month. In other words, let start by looking at the amount left to pay after the first month:

$$A_1 = P * (1 + \frac{r}{100}) - M \tag{1}$$

Let's do one more!

At the end of month two, the amount left to pay will be:

$$A_2 = A_1 * (1 + \frac{r}{100}) - M \tag{2}$$

Or:

$$A_2 = [P * (1 + \tfrac{r}{100}) - M] * (1 + \tfrac{r}{100}) - M$$

Now, a pattern might be emerging. Let's do one more month, to see if it is easier to distinguish.

At the end of month three, the amount left to pay will be:

$$A_3 = A_2 * (1 + \tfrac{r}{100}) - M$$

Or:

$$A_3 = [[P * (1 + \frac{r}{100}) - M] * (1 + \frac{r}{100}) - M] * (1 + \frac{r}{100}) - M \tag{3}$$

If you look at that equation, you will see that it can be re-arranged:

$$A_3 = P * (1 + \frac{r}{100})^3 - [M + M * (1 + \frac{r}{100}) + M * (1 + \frac{r}{100})^2] \tag{4}$$

Or:

$$A_3 = P * (1 + \frac{r}{100})^3 - M * [1 + (1 + \frac{r}{100}) + (1 + \frac{r}{100})^2] \tag{5}$$

You may do month 4 and month 5 to check, but looking at months 1 to 3, it follows that after n months, the amount left to pay will be:

$$A_n = P * (1 + \frac{r}{100})^n - M[1 + (1 + \frac{r}{100}) + (1 + \frac{r}{100})^2 + ... + (1 + \frac{r}{100})^{n-1}] \tag{6}$$

Let's take a close look at the part within square brackets:

$$1 + (1 + \frac{r}{100}) + (1 + \frac{r}{100})^2 + ... + (1 + \frac{r}{100})^{n-1} \tag{7}$$

This part is a standard geometric series with n terms.

A geometric series is the sum of numbers in a geometric progression in which the *ith* term can be defined as: $a * r^i$, where a is the first term, i is the number of term, and r is the increasing factor.

An example of this is the series:

$$2 + 10 + 50 + 250 = 2 * 5^0 + 2 * 5^1 + 2 * 5^3$$

In which 2 is the first term and 5 is the increasing factor.

To compute the total sum of the first n terms of a geometric series with $r > 1$, we can use the following formula:

$$\sum_{i=0}^{n-1} a * r^i = \frac{a(r^n - 1)}{r - 1} \tag{8}$$

So, going back to our series, we can see that the first term is 1 and the increasing factor is $(1 + \frac{r}{100})$.

By applying how to calculate the sum of the first n terms by substituting a and r as 1 and $(1 + \frac{r}{100})$, respectively, the previous equation then becomes:

$$\sum = 1 * [\frac{(1 + \frac{r}{100})^n - 1}{(1 + \frac{r}{100}) - 1}] = [(1 + \frac{r}{100})^n - 1] * (\frac{100}{r}) \tag{9}$$

Let's go back to our model, and substitute accordingly:

$$A_n = P * (1 + \frac{r}{100})^n - M[(1 + \frac{r}{100})^n - 1] * (\frac{100}{r}) \tag{10}$$

To find M, let's isolate it:

$$M = \frac{P * (1 + \frac{r}{100})^n * \frac{r}{100}}{(1 + \frac{r}{100})^n - 1} \tag{11}$$

Now that we have the mathematical model to calculate M, let's use R to solve the model quickly.

4. Solve mathematical model

Let's use R to solve the model and show how much each

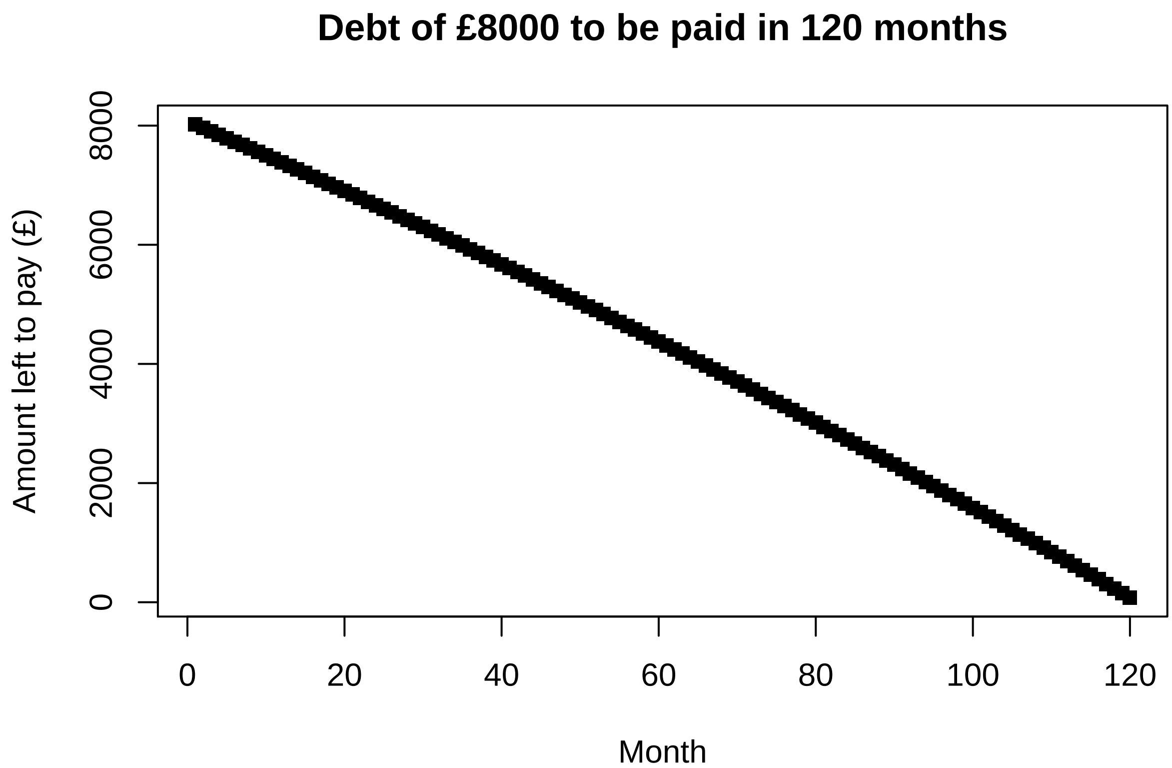
```
solve_student_loan = function(P,r,N){

  M = round(((P * (1 + r/100)^N)*(r/100))/((1 + r/100)^N -1),2)
  print(paste("The student's montly payment will be:",M))

  debt = data.frame(m = 1,
                    start = P ,
                    int = round(P*r/100,2),
                    to_pay = round(P + P*r/100,2),
```

```
                    paid = M,
                    end =  round(P + (P*r/100)-M,2)
  )

  for (n in 2:N){

    #to use the same names as before:
    A_n = debt[n-1,"end"] + debt[n-1,"end"]*r/100-M

    month = data.frame(m = n,
                       start = round(debt[n-1,"end"],2),
                       int = round(debt[n-1,"end"]*r/100,2) ,
                       to_pay = round(debt[n-1,"end"] + debt[n-1,"end"]*r/100,2),
                       paid = M,
                       end =round(A_n,2)
    )
    debt = rbind(debt,month)
  }
  #Let's show a nice graph:
  title = paste("Debt of £", P," to be paid in ", N, " months", sep="")
  plot(x = debt$m, debt$to_pay, main= title, xlab="Month", ylab= "Amount left to pay (£)",type = "l")
  points(x = debt$m, debt$to_pay,pch =15 )
  return(debt)
}
```

Let's call the model with the relevant parameters, as shown in Problem 2.

```
# Solve the model i
P = 8000
r=0.25
N = 120

ten_y = solve_student_loan(P,r,N)
```

```
## [1] "The student's montly payment will be: 77.25"
```




```
#Let's show the first few results
head(ten_y,10)
```

```
##     m   start   int  to_pay  paid     end
## 1   1 8000.00 20.00 8020.00 77.25 7942.75
## 2   2 7942.75 19.86 7962.61 77.25 7885.36
## 3   3 7885.36 19.71 7905.07 77.25 7827.82
## 4   4 7827.82 19.57 7847.39 77.25 7770.14
## 5   5 7770.14 19.43 7789.57 77.25 7712.32
## 6   6 7712.32 19.28 7731.60 77.25 7654.35
## 7   7 7654.35 19.14 7673.49 77.25 7596.24
## 8   8 7596.24 18.99 7615.23 77.25 7537.98
## 9   9 7537.98 18.84 7556.82 77.25 7479.57
## 10 10 7479.57 18.70 7498.27 77.25 7421.02
```

5. Interpret solution

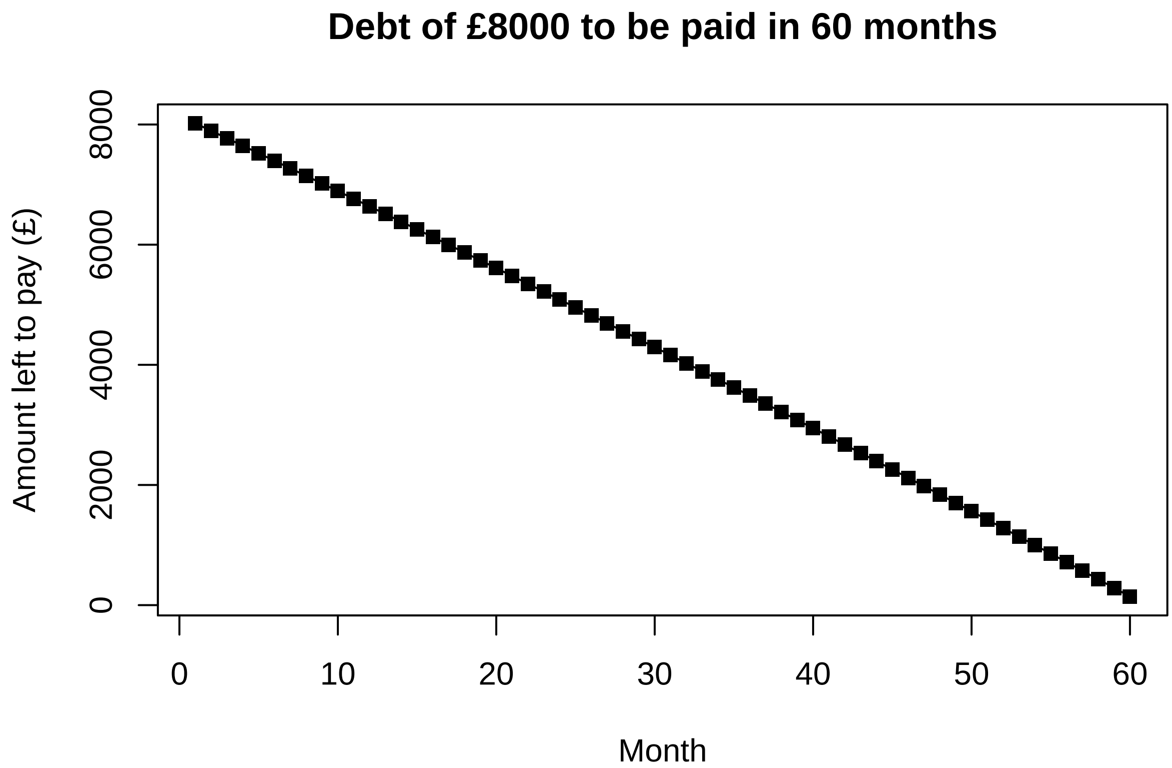
As we can see, if the student pays £77.25 per month, they'll be able to pay off their debt in 10 years.

We can also use our model to simulate several scenarios.

For example: What happens if we decrease the number of months, to 60 (i.e. giving the student 5 years to pay the loan, instead of 10)?

```r
five_y = solve_student_loan(P,r,60)
```

```
## [1] "The student's montly payment will be: 143.75"
```




```r
#let's show the first few results:
head(five_y,10)
```

```
##     m   start   int  to_pay   paid     end
## 1   1 8000.00 20.00 8020.00 143.75 7876.25
## 2   2 7876.25 19.69 7895.94 143.75 7752.19
## 3   3 7752.19 19.38 7771.57 143.75 7627.82
## 4   4 7627.82 19.07 7646.89 143.75 7503.14
## 5   5 7503.14 18.76 7521.90 143.75 7378.15
## 6   6 7378.15 18.45 7396.60 143.75 7252.85
## 7   7 7252.85 18.13 7270.98 143.75 7127.23
## 8   8 7127.23 17.82 7145.05 143.75 7001.30
## 9   9 7001.30 17.50 7018.80 143.75 6875.05
## 10 10 6875.05 17.19 6892.24 143.75 6748.49
```

Something else to consider: What happens if allow students to borrow more than £2000 a year. What happens if we let them borrow up to £5000

```r
more_funds = solve_student_loan(20000,r,N)
```

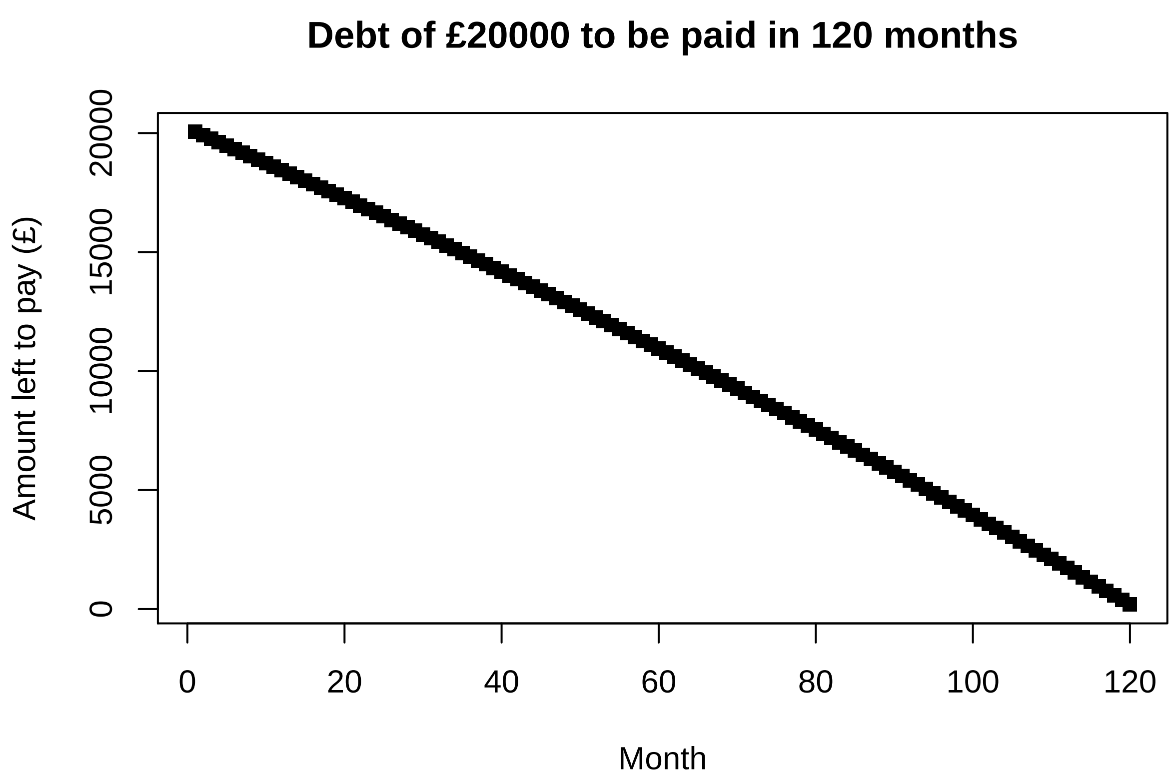
```
## [1] "The student's montly payment will be: 193.12"
```

```r
#let's show the first few results:
head(more_funds,10)
```

```
##     m    start   int   to_pay   paid      end
## 1   1 20000.00 50.00 20050.00 193.12 19856.88
## 2   2 19856.88 49.64 19906.52 193.12 19713.40
## 3   3 19713.40 49.28 19762.68 193.12 19569.56
## 4   4 19569.56 48.92 19618.48 193.12 19425.36
## 5   5 19425.36 48.56 19473.92 193.12 19280.80
## 6   6 19280.80 48.20 19329.00 193.12 19135.88
## 7   7 19135.88 47.84 19183.72 193.12 18990.60
## 8   8 18990.60 47.48 19038.08 193.12 18844.96
## 9   9 18844.96 47.11 18892.07 193.12 18698.95
## 10 10 18698.95 46.75 18745.70 193.12 18552.58
```

```r
more_funds = solve_student_loan(20000,r,120)
```

```
## [1] "The student's montly payment will be: 193.12"
```



```r
#let's show the first few results:
head(more_funds,10)
```

```
##     m    start   int   to_pay   paid      end
## 1   1 20000.00 50.00 20050.00 193.12 19856.88
## 2   2 19856.88 49.64 19906.52 193.12 19713.40
## 3   3 19713.40 49.28 19762.68 193.12 19569.56
## 4   4 19569.56 48.92 19618.48 193.12 19425.36
## 5   5 19425.36 48.56 19473.92 193.12 19280.80
## 6   6 19280.80 48.20 19329.00 193.12 19135.88
## 7   7 19135.88 47.84 19183.72 193.12 18990.60
## 8   8 18990.60 47.48 19038.08 193.12 18844.96
## 9   9 18844.96 47.11 18892.07 193.12 18698.95
## 10 10 18698.95 46.75 18745.70 193.12 18552.58
```