Workshop 7 – Solution

Customer lifetime value

MSBX-5130: Customer Analytics

3/5/2020

1) Objectives & setup

- Workshop tasks:
  1) Compute customer lifetime value (CLV) measures for a women's apparel brand
- The workshop makes use of the data file: `apparel_cust_revenue.csv` – a panel dataset containing observations of total annual revenue for a sample of 1000 customers, over a period of 10 years.
  - Sampled customers are from the same "cohort", meaning they all became customers in the same year (0).
  - Years are indexed 0 to 9 for consistency with CLV calculations on existing customers. That is, we consider the "present" (year 0) to be the period (year) in which the customer is acquired, and we calculate lifetime value with respect to this point in time.
  - When calculated in this manner, CLV represents the lifetime value of the customer (over the CLV horizon of T periods) that is realized immediately upon the acquistion of the customer.
  - CLV can also be interpreted as the maximum cost a firm should be willing to pay to acquire a customer, assuming the firm wants to break-even over the horizon of the CLV calculation.

The variables in the `apparel_cust_revenue.csv` are:

Variable	Description
`iid`	Identifier for customer
`revenue_0`	total dollars spent in year 0
`revenue_1`	total dollars spent in year 1
`revenue_2`	total dollars spent in year 2
`revenue_3`	total dollars spent in year 3
`revenue_4`	total dollars spent in year 4
`revenue_5`	total dollars spent in year 5
`revenue_6`	total dollars spent in year 6
`revenue_7`	total dollars spent in year 7
`revenue_8`	total dollars spent in year 8
`revenue_9`	total dollars spent in year 9

Workshop task workflow

1. Setup
   1. Download data & R Markdown files

    2. Load and summarize data file
    3. Functions and for loop review
2. Calculation of population-averaged CLV
    1. Simple method
        1. Determine per-customer per-period profit (M)
        2. Determine retention rate (α)
        3. Determine CLV
        4. Sensitivity to time horizon (T)
        5. Sensitivity to retention (α), interest rates (r)
    2. Cohort method
        1. Determine per-customer per-period profit dataframe/matrix (M)
        2. Determine CLV

1.1) Download data & R Markdown file

If you have not already done so, download the data file `apparel_cust_revenues.csv` from Canvas. Also download this R markdown file, `Workshop8.Rmd`.

Now launch RStudio, and change the working directory to where you have downloaded the previously mentioned files.

1.2 Load and summarize data

First, load the revenue data into a dataframe named `DF_rev`. Use `head()` and `summary()` to visualize the first few rows and to summarize the variables.

```
DF_rev = read.csv('apparel_cust_revenue.csv')
head(DF_rev)
```

```
  iid revenue_0 revenue_1 revenue_2 revenue_3 revenue_4 revenue_5 revenue_6
1  14    132.98    216.21    169.94     76.23    172.05      0.00      0.00
2  19    171.98    153.77     66.62     58.45    228.92    149.24    161.57
3  20     92.00    100.30     94.18    100.28     87.62     82.51    135.57
4  27     49.95      0.00      0.00      0.00      0.00      0.00      0.00
5  58    367.95    381.21    467.38      0.00      0.00      0.00      0.00
6  77     85.97    109.93     70.61     77.57     44.16    123.18      0.00
  revenue_7 revenue_8 revenue_9
1      0.00      0.00      0.00
2    179.18    142.36      0.00
3    117.32     52.36     95.87
4      0.00      0.00      0.00
5      0.00      0.00      0.00
6      0.00      0.00      0.00
```

```
summary(DF_rev)
```

```
      iid          revenue_0         revenue_1         revenue_2
 Min.   :   14   Min.   :   2.47   Min.   :   0.00   Min.   :   0.00
 1st Qu.: 2946   1st Qu.:  33.99   1st Qu.:  13.26   1st Qu.:   0.00
 Median : 5430   Median :  64.00   Median :  47.66   Median :  28.50
```

```
 Mean   : 5463   Mean   : 150.44   Mean   : 128.64   Mean   : 103.01
 3rd Qu.: 8110   3rd Qu.: 146.50   3rd Qu.: 113.75   3rd Qu.:  89.46
 Max.   :10589   Max.   :3135.92   Max.   :3577.04   Max.   :5456.27
   revenue_3         revenue_4         revenue_5         revenue_6
 Min.   :   0.00   Min.   :   0.00   Min.   :   0.00   Min.   :   0.000
 1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.000
 Median :  11.50   Median :   0.00   Median :   0.00   Median :   0.000
 Mean   :  78.45   Mean   :  59.19   Mean   :  46.09   Mean   :  34.942
 3rd Qu.:  66.92   3rd Qu.:  47.05   3rd Qu.:  25.24   3rd Qu.:   4.228
 Max.   :3241.10   Max.   :1822.99   Max.   :2938.26   Max.   :1662.940
   revenue_7         revenue_8         revenue_9
 Min.   :   0.00   Min.   :   0.00   Min.   :   0.00
 1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.00
 Median :   0.00   Median :   0.00   Median :   0.00
 Mean   :  26.98   Mean   :  23.68   Mean   :  19.46
 3rd Qu.:   0.00   3rd Qu.:   0.00   3rd Qu.:   0.00
 Max.   :1554.60   Max.   :1325.45   Max.   :1944.20
```

Discussion:

- What do you notice about the pattern of mean revenues over time?

Mean revenues decline rapidly over time, ranging from $150.44 in year 0 to $19.46 in year 9.

- Interpreting zero revenues as reflecting a customer being inactive, what does the pattern in the quantiles (1st, median, 3rd) over time indicate?

Over time, quantiles attain zero values, with lower quantiles attaining zeros values earlier than higher quantiles. This indicates a higher proportion of customers are becoming inactive (have zero revenues) over time.

1.2 Functions and loops review

The ability to define and use functions in R is a generally useful skill. In this workshop, we will define functions to simplify repeated calculations.

As a reminder, we can define functions using the following syntax:

```
myfunction = function(arg1, arg2, ... ){
  statements
  return(object)
  }
```

Then, calling `x = myfunction(arg1, arg2, ... )` assigns the returned object to `x`. Recall that you must declare the function in your R code before you call the function for an assignment.

Another useful construct that we will use here is the for loop. In its simplest form, the for loop spans a range of integers, where the loop index sequentially obtains the value of each integer in the range:

```
for (val in 1:5) {
  print(val)
  }
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

More generally, one can sequentially loop over an arbitrary set of numbers as in the following example:

```
val_list = c(2,5,3,9,8,11,6)
for (val in val_list) {
  print(val)
  }
```

```
[1] 2
[1] 5
[1] 3
[1] 9
[1] 8
[1] 11
[1] 6
```

2) Calculation of population-averaged CLV

2.1 Simple method

Our objective is to write a function to compute CLV under the assumption of constant customer profits and retention rates. We will call the function `CLV_simple()`.

The function should take as inputs:

- M = average profit per customer and period
- α = customer retention rate period-over-period
- r = interest/discount rate
- T = # periods for CLV horizon, so that periods run t=0:(T-1)

The output should be the estimated CLV:

$$CLV = \sum_{t=0}^{T-1} M \left(\frac{\alpha}{1+r} \right)^t$$

Before defining the function, we will first determine the inputs we derive from data: the per-customer per-period profit (M), and the retention rate (α).

2.1.1 Determine per-customer per-period profit (M)

Calculate the per-customer per-period profit (M) as the average of year 0 revenues, times the average profit margin on products sold. For this retailer, the average profit margin is approximately 40%. Assign the result to the variable M and print its value.

Note that we use the year 0 values as opposed to, for example, the average over all years because the latter would effectively "double count" the effect of customer attrition (because we explicitly account for attrition in other parts of the CLV equation).

```r
# M: customer profit/year
rmargin = 0.4
M = rmargin*mean(DF_rev$revenue_0); M
```

```
[1] 60.17796
```

2.1.2 Determine retention rate (α)

Calculate the retention rate (α) as the ratio of the number of customers who purchase (revenue>0) in year 1 to the number of customers who purchase (revenue>0) in year 0. Assign the result to the variable alpha and print its value.

```r
# alpha: retention rate
alpha = sum(DF_rev$revenue_1>0)/sum(DF_rev$revenue_0>0); alpha
```

```
[1] 0.812
```

2.1.3 Determine CLV

Now define the `CLV_simple()` function and evaluate function assuming:

- a discount rate of r = 10%
- a CLV horizon of T = 5 years (present + 4 future, indexed 0 to 4)
- per-customer per-period profit (M) and retention rate (α) as determined in 2.1.1 and 2.1.2

In your code, print output for the CLV value.

```r
# CLV_simple(M,alpha,r,T)
# M = profit per period
# alpha = retention rate per period [0,1]
# r = discount rate [0,1]
# T = # time periods for CLV calculation (0,1,...,T-1)
CLV_simple = function(M,alpha,r,T) {
  clv = 0
  for (t in 0:(T-1)) {
    clv = clv + M*(alpha/(1+r))^t
    }
  return(clv)
  }

# set discount rate, time horizon
r  = .1               # discount rate
T = 5                 # CLV horizon: 5 years total, indexed 0 to 4

# compute CLV
clv1 = CLV_simple(M,alpha,r,T); clv1
```

```
[1] 179.4668
```

Discussion:

- Based on this analysis, how much would you be willing to spend to acquire an average customer?

Based on a 5 year CLV valuation, we would be willing to spend up to $179.47. We can think of this as the 5 year break-even acquisition cost. If the firm is willing to consider longer break-even periods (e.g. 10 years), then a longer horizon can be used to evaluate willingness to pay (WTP) to acquire a customer.

2.1.4 Sensitivity to T

Demonstrate the sensitivity of CLV to different forecast horizons. Specifically, hold other factors fixed and compute CLV using your function, assuming T = {5, 10, 100, 1000} years. Print the resulting CLV values.

Hint: One approach to do this is to use a for loop with repeated calls to `CLV_simple()`.

```
Ts = c(5,10,100,1000)               # list of CLV time horizons
clv_t = rep(0,length(Ts))           # list to store CLV values
i = 1                               # integer index for CLV time horizon values

# loop over time horizon values, calculate CLV for each
for (t in Ts) {
  clv_t[i] =  CLV_simple(M,alpha,r,t)
  i = i + 1
  }

# display horizon length, CLV values in a table
disp_DF = data.frame(T=Ts,CLV=clv_t)
library(knitr)
kable(disp_DF,digits=2)
```

T	CLV
5	179.47
10	218.80
100	229.85
1000	229.85

Discussion:

- How sensitive is CLV to changes in the time horizon?

Not terribly (at these retention and discount rates) – note that at 1000 years, the CLV estimate is 229.85. At 100 years, it is the same at 2 decimal precision. At 10 years, CLV is approximately 95% of of the value at 1000 years, and at 5 years CLV is approximately 78% of of the value at 1000 years.

2.1.5 Sensitivity to retention, interest rates

Demonstrate the sensitivity of CLV to different assumptions about the retention rate (alpha) and interest rate (r). Specifically, hold other factors fixed and compute CLV using your function, forming all pair-wise combinations of the following values for alpha (retention rate) and r (interest rate): alpha = {1,.9,.8,.7}, r = {0,.05,.1,.2}. Arrange the results in a matrix (or data frame) where alpha values correspond to columns and r values correspond to rows.

Hint 1: One approach to do this is to use 2 (nested) for loops with repeated calls to `CLV_simple()`.

Hint 2: Recall that you can initialize a 2D matrix using the `matrix` *command. For example, to create a 3 row, 5 column matrix of zeros named x, use:* `x = matrix(0,nrow=3,ncol=5)`

```
# set retention rates (alpha) and interest rates (r) to evaluate
alphas = c(1,.9,.8,.7)
rs = c(0,.05,.1,.2)
# initialize the matrix that will hold CLV values
clv_ar = matrix(0,nrow=length(alphas),ncol=length(rs))

# compute CLV values
for (i in 1:length(rs)) {
  for (j in 1:length(alphas)) {
    clv_ar[i,j] =  CLV_simple(M,alphas[j],rs[i],T)
  }
}
# set matrix row/column names
colnames(clv_ar) = alphas
rownames(clv_ar) = rs

# print table of CLV values
kable(clv_ar,digits=2)
```

	1	0.9	0.8	0.7
0	300.89	246.43	202.29	166.88
0.05	273.57	226.35	187.86	156.76
0.1	250.93	209.63	175.76	148.22
0.2	215.96	183.59	156.76	134.67

```
# print table of percentage deviations from full retention/no discounting case [1,1]
kable(100*(clv_ar-clv_ar[1,1])/clv_ar[1,1],digits=2)
```

	1	0.9	0.8	0.7
0	0.00	-18.10	-32.77	-44.54
0.05	-9.08	-24.77	-37.57	-47.90
0.1	-16.60	-30.33	-41.59	-50.74
0.2	-28.23	-38.98	-47.90	-55.24

Discussion:

- How sensitive is CLV to changes in the retention rate and interest rate?

CLV is quite sensitive to changes in both retention and interest rates, with retention rates having slightly larger proportionate effects. For example, relative to the no discounting/full retention (r=0, alpha=1) scenario, increasing the discount rate by 10% reduces CLV by 16.6%, whereas decreasing the retention rate by 10% reduces CLV by 18.1%.

2.2 Cohort method

We will now write a function to compute CLV using the cohort method, and call the function `CLV_cohort()`. The function should take as inputs:

- M = a dataframe or matrix containing expected profits for each customer and time period
  - M has N rows (# customers) and T columns ("present" year 0, plus T-1 future years)
- r = interest/discount rate

Note that, consistent with the cohort method, the CLV time horizon is fixed by the number of time periods included in the profit matrix, so there is no need to supply a time horizon for the cohort method. Similarly, there is no need to specify a retention rate, as expected profits in the cohort method account for both the role of retention and profits given the customer is active.

The output should be the estimated CLV:

$$CLV = \sum_{t=0}^{T-1} M_t \left(\frac{1}{1+r}\right)^t$$

where:

$$M_t = \frac{1}{N}\sum_{i=1}^{N} M_{i,t}$$

That is, the cohort method simply takes the sample average customer profits for each time period, and discounts that average according to its year 0 present value.

Before defining the function, we will first determine the input we derive from data: the per-customer per-period profit *matrix* (M).

2.2.1 Determine per-customer per-period profit matrix (M)

For the cohort method CLV function, we want to supply as an input a matrix (or dataframe) containing expected profits for each customer and time period.

We proceed by assuming:

1. The historical pattern of customer *revenues*, as supplied in `apparel_cust_revenue.csv`, is predictive of future customer revenues.
2. We can translate between firm revenues and profits my applying the appropraite profit margin. Here, we again assume this firm has an average profit margin of 40%.

Calculate the per-customer per-period profit (M) matrix as the matrix of customer revenues (revenue-related columns in dataframe `DF_rev`) times the average profit margin on products sold. For this retailer, the average profit margin is approximately 40%. Assign the result to the variable M. Then use the `colMeans()` function to display the average customer profit by year.

```
rmargin = .4
# M: customer profit/year matrix
# (indexing removes first column in dataframe, the customer id)
M = rmargin*DF_rev[,2:dim(DF_rev)[2]]

colMeans(M)
```

```
revenue_0 revenue_1 revenue_2 revenue_3 revenue_4 revenue_5 revenue_6 revenue_7
60.177960 51.456056 41.203704 31.379960 23.677576 18.436252 13.976692 10.793484
revenue_8 revenue_9
 9.473292  7.785556
```

2.2.2 Determine CLV

Now define the `CLV_cohort()` function and evaluate function assuming:

- a discount rate of r = 10%
- per-customer per-period profits (M) as determined in 2.2.1. In your code, print output for the CLV value.

Hint: When computing CLV, be careful to properly align the values of average per-period profits (M_t) with the correct time period.

```
CLV_cohort = function(M,r) {
  # CLV_cohort(M,r)
  # M = NxT matrix/dataframe of profits (N customers over periods 0 to T-1)
  # r = discount rate [0,1]
  N = dim(M)[1]              #  # of customers
  T = dim(M)[2]              #  CLV time periods

  # compute average profits by year
  avgRev = colMeans(M)

  # compute CLV
  clv = 0
  for (t in 0:(T-1)) {
    clv = clv + avgRev[t+1]/((1+r)^t)
    # note we use avgRev[t+1] because avgRev values are indexed 1 to T, while t ranges from 0 to T-1
    }
  return(clv)
  }

# compute cohort CLV using given data, discount rate
clv2 = CLV_cohort(M,r); clv2
```

```
revenue_0
 213.3541
```

Discussion:

- How similar are the CLV estimates from the cohort method and the simple method, using the 10 year horizon?
- When might we prefer the simple CLV estimate to the cohort estimate?

The cohort estimate is (213.35-218.80)/218.80 = 2.5% lower than the (10 year horizon) simple method estimate. The estimates are thus quite close. Estimates can differ because retention rates tend to vary over time (within a single cohort). The cohort method is able to account for this time variation in retention rates. As long as the competitive environment has remained relatively stable, the cohort method is likely the more relaible CLV estimate (for a comparable time horizon) than the simple method. Conversely, if the environment has changed recently (e.g., major change in product offerings), a simple CLV calculation using the most recent 2 years of data may be the better choice.