

Water Rates from Piecewise Functions 03-19

Note : this pdf note was generated directly from a *Mathematica* notebook. [rob r, 2008-03-15, 19]

This exercise was occasioned by one of my colleagues (Tom Gaskill) describing some water usage data and ideas about its' analysis. It looked like the water usage was billed by a procedure that had stepwise increments as water usage increased. This is just like price breaks in purchasing. For example, the first 100 items are a certain price while those over 100 are priced at a reduced rate while those over 1000 are priced even lower. The function could go the other direction as well though, as a commodity or service became more scarce/valuable in larger volumes (like space on a life boat or luggage weight on an airline!). So, I was playing with some *Mathematica* functions to try to express this. See the provider[. .] function below as well as plots of cost curves associated with its use.

Note : There is an alternative way to show these piecewise functions. An example is at the end of this tutorial.

Once I have the provider function, I will show a few plots of the cost curve for some made-up data.

For example, for provider1, say the 'fixed' cost is 2 units, which is the cost before the tap is even opened.

I use small values, but of course, these can be scaled. Then, suppose the price $p1$ for the first 5000 gallons is 7 units. In the provider function, I also pro-rated the cost to be: $7 * \text{gallons of usage} / 5000$ in case someone used less than 5000.

So the cost for using all 5000 gallons would be $2 + 7 = 9$. If less than 5000, I pro-rated it as above.

Then, suppose the next block of water is priced at say, $p2 = 3$ units for a block, say

between 5000 and 10000. For this block of water the cost would be

fixed + $p1 + p2 * (\text{gallons} - 5000) / (10000 - 5000)$, So if someone used 8000, then I would calc this as :

fixed + $p1 + p2 * 3/8$

fixed = \$ fixed cost of services from the provider

gal = cumulative gallons of usage (It's best to use low numbers here and scale)

$p1, p2, p3, \dots$ price/block specific to blocks 1, 2, 3, . . .

where a 'block' means a range of water allocation like

between 5000 and 10000 gallons

If the price for that block is \$10, then the cost is \$10/5000 gallons

or \$1/500 per gallon within that range

The plan here is to represent the providers rates per cumulative gallons used , specific to each block of water allocation.

Basic Provider Piecewise Function

This function takes an identifier of the water provider,

id = provider identifier

gal = gallons used, the variable whose value determines the cost.

fixed = fixed cost before any water is actually delivered

blocks - a list of 3 component lists, where each component is

{ p, lo, hi } where p is the block cost, lo is the extreme lower value of the water block while hi is the high extreme value, e.g.

blocks = { { 7, 0, 5000}, {4, 5000, 10000}, { 1, 10000, 18000} }

For example, block 2 has a price of 4, and the range is between 5000 and 10000.

Just as an aside, it's best to use small numbers and scale later

```
Clear[provider, a, b, c, s, t, u, v, b1, b2, temp, arg, numberBlocks, blocks, p, lo, hi, i]

provider [id_, gal_, fixed_, blocks_] :=
Module[{price, temp, arg, numberBlocks, p, lo, hi },
  numberBlocks = Length[blocks];
  Print["numberBlocks  ", numberBlocks];
  price = fixed;
  arg = {}; (* start the build vector of args for Piecewise *)
  For[i = 1, i ≤ numberBlocks, i++,
    p = blocks[[i, 1]]; (* block price *)
    lo = blocks[[i, 2]]; (* low range value of block water usage*)
    hi = blocks[[i, 3]]; (* high range value of block water usage *)
    temp = {price + p * (gal - lo) / (hi - lo),
      lo < gal < hi}; (* low to high interval *)
    arg = Append[arg, temp]; (*build up Piecewise arguments *)
    (*Print["i ", i, " price " ,price, " ",arg]; *)
    price = price + p; (* set up for next iteration *)
  ]; (* end For loop *)
  Piecewise[arg] (*load arguments to piecewise function *)
]
```

■ ***** End Function *****

Provider 1 Data

■ **Note: **** This is strictly hypothetical data made up to play with**

Just to start, I am using just three blocks of water rates, (the program will handle any number of blocks).

fixed price = 3

block 1 :

price = 3

low range = 0, high range = 5 (maybe scale by 1000?)

block 2:

price = 2

low = 5, high = 10

block 3:

price = 1.5

low = 10 high = 25

Note that the blocks are lists and are enclosed in a list. That way, I can tell how many blocks there are... The function displays its interior structure as a piecewise function. So, for gallon usage between 0 and 5, I would bill the customer at \$3 fixed charge, plus prorated cost from 0-5 with a cost of \$3 * (gallons - 5000)/(5000 -0). A total of \$6 for 5000 units of water usage.

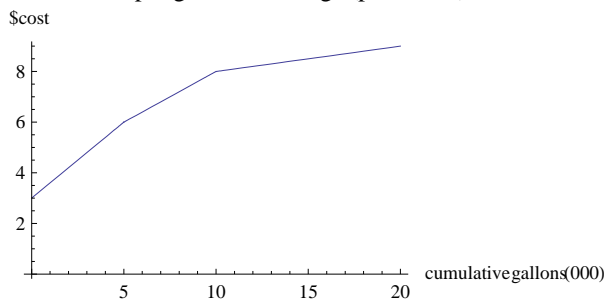
```
provider1 = provider["Joe", gal, 3,
  {{3, 0, 5}, {2, 5, 10}, {1.5, 10, 25}}]
```

```
numberBlocks 3
```

$$\begin{cases} 3 + \frac{3 \text{ gal}}{5} & 0 < \text{gal} < 5 \\ 6 + \frac{2}{5} (-5 + \text{gal}) & 5 < \text{gal} < 10 \\ 8 + 0.1 (-10 + \text{gal}) & 10 < \text{gal} < 25 \end{cases}$$

```
p1 = Plot[provider1, {gal, 0, 20},
  PlotLabel -> "Provider 1 Cost per gallons of usage (pro-rated)", AxesOrigin -> {0, 0},
  AxesLabel -> {"cumulative gallons(000)", "$cost "}]
```

Provider 1 Cost per gallons of usage (pro-rated)



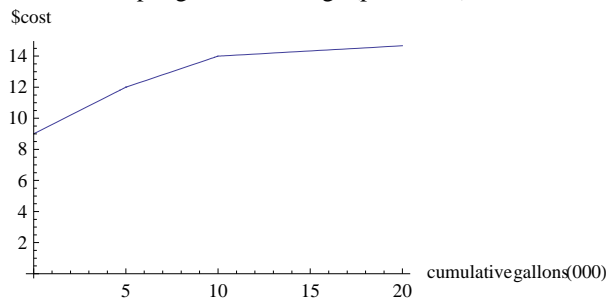
```
provider2 = provider["Sammi", gal, 9,
  {{3, 0, 5}, {2, 5, 10}, {1.0, 10, 25}}]
```

```
numberBlocks 3
```

$$\begin{cases} 9 + \frac{3 \text{ gal}}{5} & 0 < \text{gal} < 5 \\ 12 + \frac{2}{5} (-5 + \text{gal}) & 5 < \text{gal} < 10 \\ 14 + 0.0666667 (-10 + \text{gal}) & 10 < \text{gal} < 25 \end{cases}$$

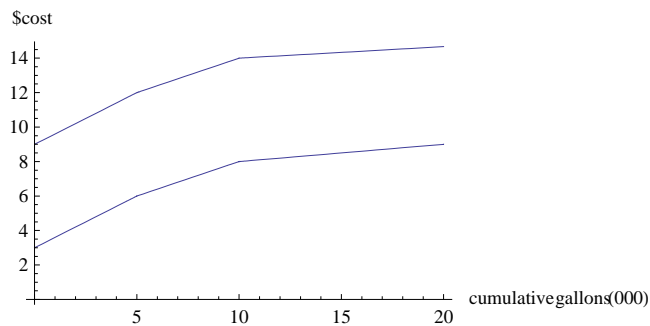
```
p2 = Plot[provider2, {gal, 0, 20},
  PlotLabel -> "Provider 2 Cost per gallons of usage (pro-rated)", AxesOrigin -> {0, 0},
  AxesLabel -> {"cumulative gallons(000)", "$cost "}]
```

Provider 2 Cost per gallons of usage (pro-rated)



```
Show[p1, p2, PlotRange -> Automatic, PlotLabel -> "Provider 1 & 2"]
```

Provider 1 & 2



■ An Alternative Piecewise Description

Mathematica has built in function called `Piecewise`, that I used above, that will let me enter functional values on a per interval basis. The approach below is more verbose than the *Piecewise* function, but will allow a bit more variation.

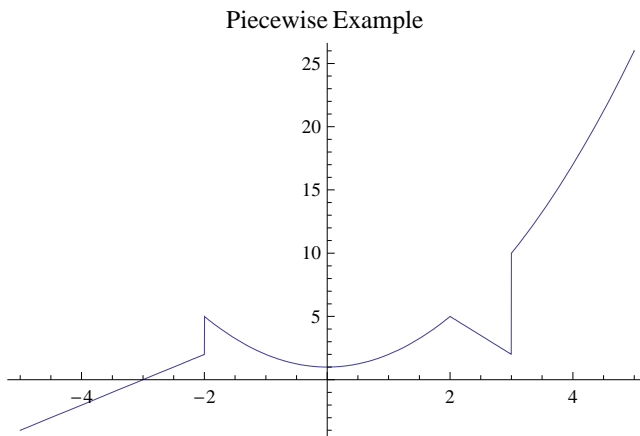
```
Clear[h, x]

h[x_] := x^2 + 1 /; -2 < x ≤ 2 || x > 3 (* define h over disjoint intervals*)

h[x_] := 11 - 3 x /; 2 < x < 3

h[x_] := 6 + 2 x /; x ≤ -2
```

```
Plot[h[x], {x, -5, 5}, PlotLabel -> "Piecewise Example ",  
AxesOrigin -> {0, 0}]
```



Summary

The application here is just a toy example but the math is not trivial. Further work could include various tables generated for different scenarios and additional function can be fairly easily defined. *Mathematica* is extensible in limitless way, if I just knew enough!