test

Diego Aviles

Wednesday, July 29, 2015

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Part 2:

Attribute number 1 in our `camera_data` is the digital camera's resolution. Consider the following: you want to research how the importance of the resolution attribute correlates with the prospect value of the chosen camera. Remember that the last round in the end configuration represents the specifications of the camera the user decided to choose. As the 'importance of the resolution attribute' we will use the relative sum of the values for the resolution category, throughout all rounds, for each user. Assuming the package is loaded, we first must extract the decision matrix for all users, but only for the first attribute, i.e. the resolution.

```
x1 <- powerful_function(camera_data, userid= get_all_userids(camera_data),
                        FUN= decision_matrix,
                        attr = 1,
                        rounds="all")
```

```
tail(x1, n= 4)
```

```
## $usid60
##        attr1
## round0     1
## round1     1
## round2     1
## round3     3
## round4     2
## round5     2
##
## $usid61
##         attr1
## round0      2
## round1      2
## round2      1
## round3      1
## round4      1
## round5      1
## round6      1
## round7      2
## round8      2
## round9      2
## round10     2
## round11     2
## round12     1
## round13     1
## round14     1
## round15     2
##
```

```
## $usid62
##        attr1
## round0     1
## round1     1
## round2     1
##
## $usid63
##        attr1
## round0     1
```

Then, we need to (1)sum all values for `attr1` for each user, (2) calculate how many rounds each user has and (3) divide the sum of the values with the total amount of rounds, for each user. The value of `x` is the one we are going to use as the x-axis value for our plot.

```
x2 <- sapply(x1, sum)     ## sum gains (1)
x3 <- sapply(x1, length)  ## amount of rounds (2)
x <- x2/x3                ## relative gains
```

```
tail(x)
```

```
##    usid58    usid59    usid60    usid61    usid62    usid63
## 1.2500000 0.8888889 1.6666667 1.5000000 1.0000000 1.0000000
```

Further, using only one the `powerful_function` together with the `overall_pv` in 'productConfig', the desired overall prospect values are obtained.

```
y <- powerful_function(camera_data, userid= get_all_userids(camera_data),
                       FUN=overall_pv,
                       rounds= "last",
                       cost_ids= 4,
                       attr= c(1,2,3,4))
y
```

```
##      usid6      usid9     usid10     usid11     usid12     usid13
## -1.3552678  0.5676987 -0.5361368 -0.1306124  0.3312160  0.5632730
##     usid14     usid15     usid16     usid17     usid18     usid19
##  0.3312160  0.9486505  0.5150779  0.5703114  0.5756948  0.5804537
##     usid20     usid21     usid22     usid25     usid26     usid27
##  0.2781216 -0.6905374  0.7981669  0.4984120  0.5650541  0.4984120
##     usid28     usid29     usid30     usid31     usid32     usid33
##  0.0000000  0.0000000  0.0000000  0.5163688  0.5541440  0.4984120
##     usid34     usid35     usid36     usid37     usid38     usid39
##  0.0000000  0.0000000  0.5804537  0.4798347  0.5169893  0.1137083
##     usid40     usid41     usid42     usid43     usid44     usid45
## -1.2085092  0.5541440 -0.8886093  0.0000000  0.7838494  0.0000000
##     usid46     usid47     usid48     usid49     usid50     usid52
##  0.5541440  1.0000000  0.5262780  0.4984120  0.0000000  0.4984120
##     usid53     usid54     usid57     usid58     usid59     usid60
##  0.4176973  0.0000000  0.6992180  0.4984120  0.5136002  0.6360608
##     usid61     usid62     usid63
##  0.4148140  0.0000000  0.0000000
```

At this point cost_ids. . . . . To add another level of complexity to our analysis, assume you know need to include the influence of the amount of interaction the users had with the camera configurator. We can quantify this by using the amount of total clicks for each user. The more rounds a user has, the more he interacted with the configurator. The results can be obtained by inputting the `get_rounds_by_ID` in our `powerful_function`. As a second step, we use `sapply` to simplify the result from a list to a vector.

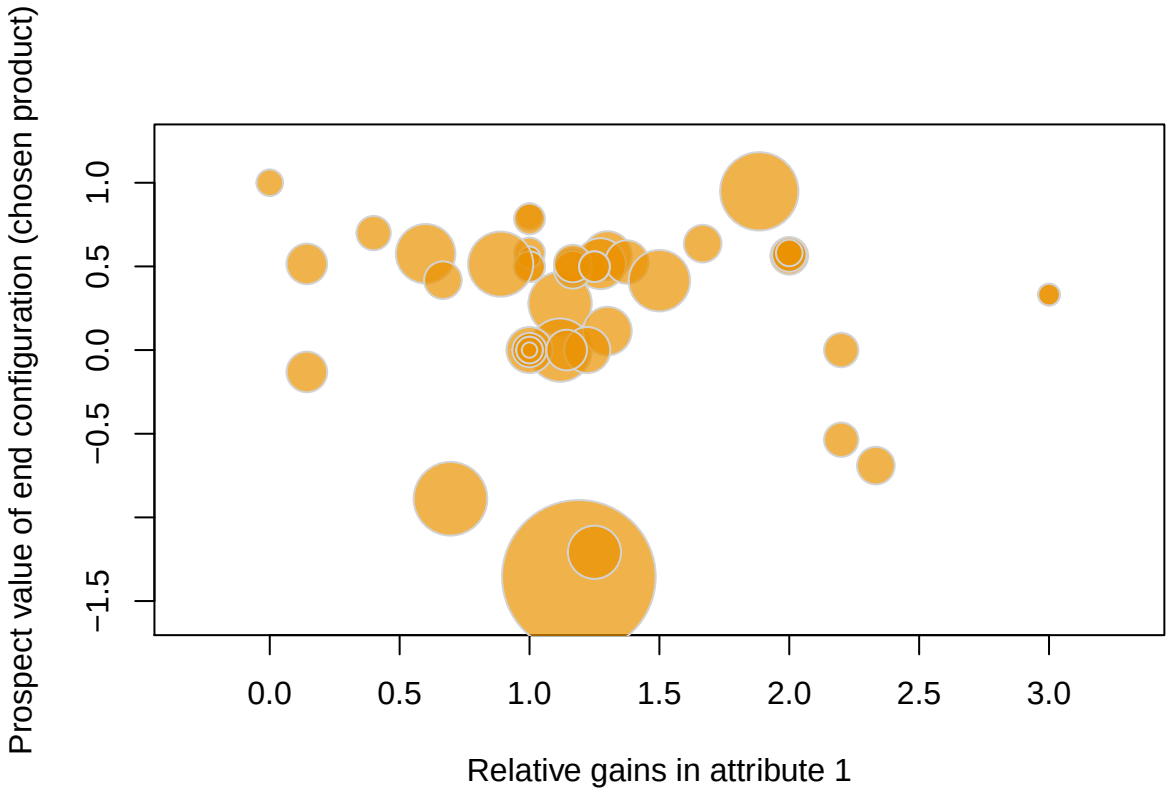
```
z1 <- powerful_function(camera_data, userid= get_all_userids(camera_data),
                        FUN=get_rounds_by_ID)
z <- sapply(z1, length)  ## amount of clicks for all users

z
```

```
##  usid6  usid9 usid10 usid11 usid12 usid13 usid14 usid15 usid16 usid17
##    100      4      5      7      2      6      2     26      7      5
## usid18 usid19 usid20 usid21 usid22 usid25 usid26 usid27 usid28 usid29
##     15      3     17      6      3      4     10      4      5      5
## usid30 usid31 usid32 usid33 usid34 usid35 usid36 usid37 usid38 usid39
##     17     11      2      6      9      5      4      6      6     10
## usid40 usid41 usid42 usid43 usid44 usid45 usid46 usid47 usid48 usid49
##     12      2     23      9      4      5      2      3      8      4
## usid50 usid52 usid53 usid54 usid57 usid58 usid59 usid60 usid61 usid62
##      3      4      6      7      5      4     18      6     16      3
## usid63
##      1
```

Using the `powerful_function` in combination with other functions from our package and other 'R' auxiliary operations, we were able to mine the desired data. The values for `x, y, z` alone are difficult to analyze. 'R' offers different grapichal solutions for plotting data. Even though, plotting is outside of the scope of this package, we want to give an example so that the values for `x, y, z` can be better appreciated. For this purpose we use the `symbols` function from the `graphics` base-package from 'R' [16-17].

```
symbols(x, y, circles= sqrt(z /pi),
        inches= 0.40, fg="lightgray",
        bg= rgb(235, 146, 1, alpha= 180, maxColorValue= 255),
        xlab="Relative gains in attribute 1",
        ylab="Prospect value of end configuration (chosen product)")
```


Prospect value of end configuration (chosen product)
1.0
0.5
0.0
−0.5
−1.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Relative gains in attribute 1