Treatment/Dosage Spells Algorithm

author: “Aaron D. Schroeder”
date: “10/23/2014”
description: Takes treatment data that is in long-format, transforms and casts the data to wide-format, then finds all consecutive-month treatment runs (spells).

1. Add Libraries and Import Data

Sample data includes client id and service/treatment date.

```
library(reshape2)                                                    # Load reshape2 library
reshapeData <- read.csv("R dosage test data.csv")                    # Import csv data
print(reshapeData)
```

```
##    clientid servicedate
## 1  Client A      1/1/14
## 2  Client A      2/1/14
## 3  Client A      3/1/14
## 4  Client A      5/1/14
## 5  Client B      2/1/14
## 6  Client B      3/1/14
## 7  Client B      4/1/14
## 8  Client B      6/1/14
## 9  Client B      7/1/14
## 10 Client C      1/1/14
## 11 Client C      2/1/14
## 12 Client C      5/1/14
## 13 Client C      7/1/14
```

2. Create New Transposed Data Frame

Transform service date into month_year. Add a received column to be used in the cast step to show months receiving service.

```
clientid <- reshapeData$clientid                                     # Create vectors for new df
monthYear <- format(as.Date(reshapeData$servicedate, "%m/%d/%y"), "%m_%Y")
received <- ifelse(!is.null(reshapeData$servicedate), 1, 0)
newDF <- data.frame(clientid, monthYear, received)                           # Create data frame
print(newDF)
```

```
##    clientid monthYear received
## 1  Client A   01_2014        1
## 2  Client A   02_2014        1
## 3  Client A   03_2014        1
## 4  Client A   05_2014        1
## 5  Client B   02_2014        1
## 6  Client B   03_2014        1
## 7  Client B   04_2014        1
## 8  Client B   06_2014        1
## 9  Client B   07_2014        1
## 10 Client C   01_2014        1
## 11 Client C   02_2014        1
## 12 Client C   05_2014        1
## 13 Client C   07_2014        1
```

3. Cast to wide data frame

Creates a single row per client id with “1” indicating service received, otherwise “0”.

```
castDF <- dcast(newDF, clientid ~ monthYear, value.var="received", fill="0") # Reshape df with dcast
print(castDF)
```

```
## 1 Client A       1       1       1       0       1       0       0
## 2 Client B       0       1       1       1       0       1       1
## 3 Client C       1       1       0       0       1       0       1
```

4. Build data frame to be filled

Include columns for spell count, spell lengths (a comma delimited string of spell lengths), max spell length, and min spell length.

```
vars <- c("spell count", "spell lengths (mos)", "max spell length", "min spell length")
finalDF <- data.frame(vars)                                          # Add vector (column) to df
```

5. Get all runs (dosage spells) per id

Loops through wide data frame and uses the R Run Length Encoding (RLE) function to get runs of consecutive integers (in this case "1"). Would be faster using vectorization but looping is easier to understand and generally fast enough.

```
for (i in 1:NROW(castDF) ) {
  currentRow <- castDF[i,]                                           # Get row i from wide data frame
  print(currentRow)
  currentId <- currentRow[[1]]                                       # Get client id from row i
  print(currentId)
  runs <- rle(currentRow)                                            # Use RLE to get run lengths
  lengths <- sort(as.character(runs$length[runs$values == 1]))       # Get and sort runs with value 1
  print(lengths)
  numspells <- length(lengths)                                       # Length of vector
  maxspell <- tail(lengths, n=1)                                     # Last item in vector
  minspell <- lengths[1]                                             # First item in vector
  finalDF[as.character(currentId)] <- NA                             # Create new df column
  newVector <- c(numspells, toString(lengths), maxspell, minspell)   # Create new values vector
  finalDF[[as.character(currentId)]] <- newVector                     # Add vector to finalDF
}
```

```
##   clientid 01_2014 02_2014 03_2014 04_2014 05_2014 06_2014 07_2014
## 1 Client A       1       1       1       0       1       0       0
## [1] Client A
## Levels: Client A Client B Client C
## [1] "1" "3"
##   clientid 01_2014 02_2014 03_2014 04_2014 05_2014 06_2014 07_2014
## 2 Client B       0       1       1       1       0       1       1
## [1] Client B
## Levels: Client A Client B Client C
## [1] "2" "3"
##   clientid 01_2014 02_2014 03_2014 04_2014 05_2014 06_2014 07_2014
## 3 Client C       1       1       0       0       1       0       1
## [1] Client C
## Levels: Client A Client B Client C
## [1] "1" "1" "2"
```

Final dosage spells data frame

```
print(finalDF)
```

```
##                 vars Client A Client B Client C
## 1         spell count        2        2        3
## 2 spell lengths (mos)     1, 3     2, 3  1, 1, 2
## 3    max spell length        3        3        2
## 4    min spell length        1        2        1
```