Processing Observation Data from Learning Experiments

Lewis Marsh

06/06/2018

This R Markdown report aims to summarise the R script `processObservationData.R`. It consists of two main functions, `processObservationData` and `processAllObservationData`.

Processing a Single Data Set

The first of these two functions,

```
processObservationData(data,
                       date=as.POSIXct(strptime("10-01-2018", "%d-%m-%Y")),
                       project="Isle",
                       activitycol=F,
                       observercol=F,
                       namecols=c("timestamp","group",
                            "StudentA","StudentB","StudentC","StudentD")),
```

takes in a dataframe together with some other parameters and returns a cleaned data frame.

In the orginal data sheets, each entry lists the observation for a whole group of students (between 3 and 6 usually) and the observations are listed as a piece of (predefined) text, such as "Talking with their group peers to solve the task" or "Totally disengaged".

In the output this is broken up into individual entries for each student, for which a student variable is introduced. Furthermore, each possible observation is turned into a boolean variable for each of these entries, which makes it easier to conduct further statistical analysis of the dataset. Also, a column with additional notes from the original dataframe is dropped in the output, as it is difficult to make use of in a statistical test, and a global ID is created for each student, using the date, group, student number in the group and the project name. This will be of use when merging several datasets with the second function.

As students undertook several tastks in some of the experiments/projects, the input variable "activitycol" indicates whether the function has to look for an extra activity column in the input data. This will be taken over one-to-one in the output or will be filled in with "Standard" if there was only one activity. Similarly, in some projects there were more than one observer placed onto each group to enable to test the reliability of the observers and the data they produce. The variable "observercol" tells the function whether there is a separate observer column to take over from the input data. Otherwise "1-A" is filled in for each observer.

Processing and Merging of Several Data Sets

The second function, `processAllObservationData` takes in a vector of URLs in which the GoogleDocs containing the experimental data are stored in. It reads the data from the online sheets and places them in a dataframe. By examining the names of the columns in the dataframe, it detects whether there are activity or observation columns. By inspecting the first time stamp, it determines the date of the experiment. It then cleans the data using the first function and adds it to the end of a larger dataframe which eventually gets returned after the function has iterated through all URLs.

Example of Output

To exhibit what the functions do, there is a brief example. We are given a data frame taken from one of the GoogleDocs:

```
raw_data[1,1:2]
```

```
##             Timestamp What.group.are.you.observing.
## 1 11/10/2017 10:18:07                       Group 1
```

```
raw_data[1,3]
```

```
## [1] "Talking with their group peers to solve the task"
```

These are only two snippets of the data, as anything else would not fit onto this page. We see that the first two columns give a timestamp and the group. We furthermore see the oberservation for student A. There are 3 other students in the group with individual columns as well as a column for further comments.

By applying the `processObservationData` function, we then see how the student variable is created and the observations are turned from text into boolean values. Furthermore, the activity and observer column have been created which makes it easier to merge and compare this data with other datasets.

```
processed_data <- processObservationData(raw_data)
summary(processed_data)
```

```
##   timestamp            group               student      disengaged
##  Length:1085        Length:1085        StudentA:155   Min.   :0.00000
##  Class :character   Class :character   StudentB:155   1st Qu.:0.00000
##  Mode  :character   Mode  :character   StudentC:155   Median :0.00000
##                                        StudentD:155   Mean   :0.07189
##                                        NA      :155   3rd Qu.:0.00000
##                                        NA      :155   Max.   :1.00000
##                                        NA      :155
##     looking          talking         technology      resources
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.000
##  Median :0.0000   Median :1.0000   Median :0.000   Median :0.000
##  Mean   :0.2258   Mean   :0.5991   Mean   :0.388   Mean   :0.388
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.000
##
##     external       student.id          activity           observer
##  Min.   :0.0000   Length:1085        Length:1085        Length:1085
##  1st Qu.:0.0000   Class :character   Class :character   Class :character
##  Median :0.0000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.1539
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
##    project               date             global.id
##  Length:1085        Min.   :2018-01-10   Length:1085
##  Class :character   1st Qu.:2018-01-10   Class :character
##  Mode  :character   Median :2018-01-10   Mode  :character
##                     Mean   :2018-01-10
##                     3rd Qu.:2018-01-10
##                     Max.   :2018-01-10
##
```

```r
str(processed_data)
```

```
## 'data.frame':    1085 obs. of  15 variables:
##  $ timestamp : chr  "11/10/2017 10:18:07" "11/10/2017 10:25:51" "11/10/2017 10:28:37" "11/10/2017 10
##  $ group     : chr  "Group 1" "Group 1" "Group 1" "Group 1" ...
##  $ student   : Factor w/ 7 levels "StudentA","StudentB",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ disengaged: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ looking   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ talking   : num  1 1 0 1 1 1 0 0 0 0 ...
##  $ technology: num  0 1 1 1 0 0 1 1 1 0 ...
##  $ resources : num  0 1 1 1 0 0 1 1 1 0 ...
##  $ external  : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ student.id: chr  "Group 1 StudentA" "Group 1 StudentA" "Group 1 StudentA" "Group 1 StudentA" ...
##  $ activity  : chr  "Standard" "Standard" "Standard" "Standard" ...
##  $ observer  : chr  "1-A" "1-A" "1-A" "1-A" ...
##  $ project   : chr  "Isle" "Isle" "Isle" "Isle" ...
##  $ date      : POSIXct, format: "2018-01-10" "2018-01-10" ...
##  $ global.id : chr  "Isle 2018-01-10 Group 1 StudentA" "Isle 2018-01-10 Group 1 StudentA" "Isle 2018
```