DATA PRE-PROCESSING AND CLEANING HOMEWORK

Instructions

For each question report the R code and the obtained result in Markdown file called `data_clean.md` or `data_clean.Rmd` inside a data_cleaning folder of your github repo (you can freely choose to use an already existent repo in you github or to create a new one).

- Once you complete your Exercices, send me the link of the file to matteo.manca@eurecat.org
- ***The deadline is March 14, 2016***
- If you do not understand something in the exercises or questions, please send me an e-mail (also for easy doubts, no problems!!)
- Since we are working in R, ***I would suggest to use the Rmd format:*** Rmd files are R Markdown files (so you can read and write them using RStudio) that enable easy creation of dynamic documents, presentations, and reports from R (so, from Rstudio). They are very convenient since it is possible to execute R code within this file. To see an example of Rmd file download .Rmd Example.

PS: To see the output of .Rmd file in RStudio, click on the "Knit HTML" (to generate an html as output) or "Knit PDF" (to generate a PDF as output)

Introduction

This assignment uses two twitter datasets available at the following links:

- https://www.dropbox.com/s/20apq0yj75evvtl/user_info.csv?dl=0
- https://www.dropbox.com/s/tjh5t5b72hni8qm/tweet_info.csv?dl=0

Let's see more details about the datasets.

The file ***user_info.csv*** contains the following data:

1. ***tweet_id:*** The unique identifier referring to a Tweet
2. ***user_id:*** The unique identifier referring to the user that post the corrispondent tweet
3. ***Localtion:*** location of the user 4: ***Tourist:*** it takes value 1 if the user is a tourist, 0 otherwise

The file ***tweet_info.csv*** contains the following data:

1. ***tweet_id:*** The unique identifier referring to a Tweet
2. ***timestamp:*** Date and time in which the tweet has been posted in the format "yyyy-mm-ddTHH:MM:SS"
3. ***latitude:*** latitude
4. ***longitude:*** longitude

Download data and save them in your working directory

- https://www.dropbox.com/s/20apq0yj75evvtl/user_info.csv?dl=0
- https://www.dropbox.com/s/tjh5t5b72hni8qm/tweet_info.csv?dl=0

Load the datasets into two different data frames using R

1. load “user_info.csv” in a dataframe called “df_user”
2. load “tweet_info.csv” in a dataframe called “df_tweets”

Consider that:

- the downloaded files DOES NOT contain the header
- NA values are represented by the “NOT-AVAILABLE” string
- the separator character is the “,”.

Data Pre-processing and cleaning

Ex1. Explore the datasets by showing first lines and the *result of various model fitting functions*

Ex2.Assign clear names to each column of the two datasets and show a preview of the first lines (the names could be the same listed in the introduction for instance) If the values that correspond to the tweet_id are in a “unusual” format, run the following command to encode tweet_ids as a factor

```r
df_tweets$tweet_id <- factor(df_tweets$tweet_id)
df_user$tweet_id <- factor(df_user$tweet_id)
head(df_tweets)
```

```
##             tweet_id           timestamp latitude longitude
## 1 603928128508010496 2015-05-28T16:17:40 41.40235  2.188129
## 2 572767011664748544 2015-03-03T15:34:31 41.35630  2.131096
## 3 572768427426910208 2015-03-03T15:40:09 41.35630  2.131441
## 4 572817210160357376 2015-03-03T18:53:59 41.35621  2.130454
## 5 612574920053248000 2015-06-21T12:56:56 41.38121  2.183924
## 6 612576567148023808 2015-06-21T13:03:29 41.37801  2.184717
```

```r
head(df_user)
```

```
##             tweet_id user_id                  location turist
## 1 603928128508010496     521        Amsterdam  Seattle      1
## 2 572767011664748544    1866                      <NA>      1
## 3 572768427426910208    1866                      <NA>      1
## 4 572817210160357376    1866                      <NA>      1
## 5 612574920053248000    1966 San Francisco  California      1
## 6 612576567148023808    1966 San Francisco  California      1
```

Conceptually, factors are variables in R which take on a limited number of different values; such variables are often refered to as categorical variables" (see http://www.stat.berkeley.edu/~s133/factors.html for more details).

Ex3. What are the dataset dimensions?

Ex4. How many DISTINCT locations there are in the df_user data frame ?

Ex5. How many DISTINCT users there are in the df_user data frame ?

Ex6. How many DISTINCT tweets there are in the df_user data frame ?

Ex7. Check for duplicates in both datasets. If there are duplicate rows in the data frame do the following steps (repeat these steps for both dataframes):

1. count duplicates in each dataset
2. extract the duplicate rows and save them in a new temporary data frame;
3. save the temporary data frame ina .csv file in your working directory
4. remove the duplicate rows from the original dataframe
5. check the dimensions of the new dataframes without duplicates

Ex8. Merge the two dataframes in a unique dataframe “twitter_df”(note that they share a common field)

Ex9. From “twitter_df” dataframe, create two dataframes:

1. tour_df containing onlty tourists data
2. local_df containing onlty no-tourists (locals) data

Ex10. Write the dataframe just created into two separated files

Ex11. create a new dataset “no_complete_twitter_df” containg all rows of “twitter_df”" dataset that are not completed (i.e., that contain NA values). Show a preview of the just created dataset. How many rows does the new dataset (no_complete_twitter_df) have?

Ex12. Extract month and hour from the timestamp field and add them to the dataframe as separate columns :

1. month
2. hour PS: The functions *as.POSIXlt* and *as.POSIXlt* can be useful!

Ex13. Compute the number of tweets per month discerning tourists and locals (no-tourists) Save the result in a new df

Ex14. From “twitter_x_mont” create a new dataframe to avoid to repeat each date twice (one time for turists and one time for locals) [OPTIONAL] Use ***reshape*** function: This function reshapes a data frame between ‘wide’ format with repeated measurements in separate columns of the same record and ‘long’ format with the repeated measurements in separate records.

The resulting dataframe will appear as follow:

```
##     month tweet_id.0 tweet_id.1
## 1 2015-01      91410      11870
## 2 2015-02      68284      12556
## 3 2015-03      51435      18140
## 4 2015-04      54276      10487
## 5 2015-05      18853       7044
```

Where `tweet_id.0` represents the number of locals and `tweet_id.1` represents the number of tourists

Ex15. Create a new dataset "tweets_x_user" containing the number of tweets for each user_id
PS: Use aggregate and rename the field that contains the twitter count

Ex16. Make some check related to the previous point (i.e., select all tweets for a given user from twitter_df and verify if the number of tweets corresponds)

Ex17. Sort the "tweets_x_user dataset" to put on the top user with a higher number of tweets and print the user_id with the higher number of tweet