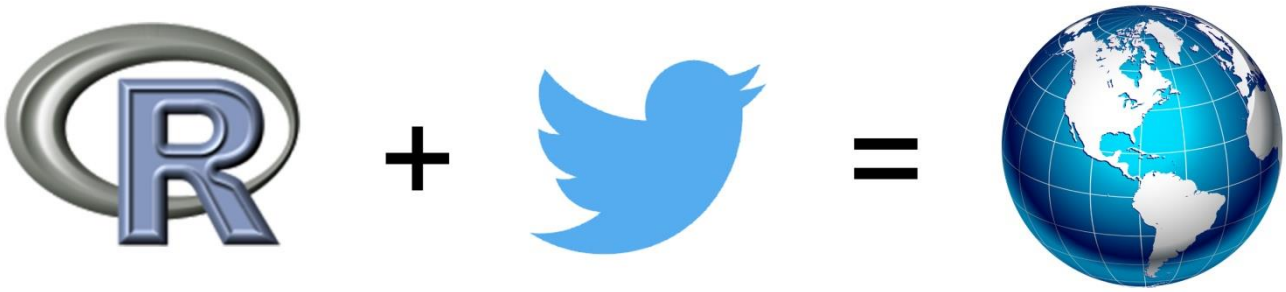
SOFTWARE REQUIREMENT SPECIFICATIONS

The following project aims at creating a software that can analyze Twitter posts and return a description of the social and political situation of the world.

The idea behind the project is that the wealth of a country can be estimated just knowing how some topic (like war, political elections, religion, economic crisis) are felt among its citizens.
If a topic is well present in a country, the citizens will create tweets that refer to that topic. Thus the number of tweets that treat a certain topic is directly proportional to its importance.

The software must be developed using R language and it must work in Rstudio.

The libraries that should be used are *TwitteR* in order to create a connection with the website's API and the package *tm* to further analyze the words of the tweets once imported in Rstudio.



The final output of the software will be an array that contains, for each day and for each state, a score indicating how a topic is worth.

The image below is purely indicative but depict the logic behind the output file and its structure.

		Afghanistan			Albania			Algeria			Algeria			Argentina			Australia		
date	day	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections
01/01/2014	1	76	0	98	16	30	63	67	72	95	85	76	21	0	19	57	44	95	40
02/01/2014	2	15	4	54	61	49	46	46	76	98	25	32	76	1	45	46	12	93	40
03/01/2014	3	25	41	55	5	8	61	98	41	79	33	51	58	66	8	55	98	60	81
04/01/2014	4	51	74	71	67	82	79	55	33	37	68	61	70	88	63	90	15	33	80
05/01/2014	5	74	17	15	27	7	46	37	91	8	54	88	27	12	80	85	75	50	78
06/01/2014	6	90	27	2	79	43	80	62	41	41	74	91	89	60	58	97	73	92	71
07/01/2014	7	16	96	61	18	77	46	9	32	11	0	38	83	81	76	82	26	24	64
08/01/2014	8	13	10	3	17	28	3	23	17	63	90	93	36	40	2	81	37	66	16
09/01/2014	9	31	10	66	48	8	20	50	45	41	20	90	73	98	32	60	69	31	16
10/01/2014	10	52	29	82	86	61	49	62	28	4	38	88	90	12	58	6	44	26	89
11/01/2014	11	28	43	57	16	35	43	26	25	65	48	20	77	4	52	33	31	79	7
12/01/2014	12	63	67	41	38	50	52	77	99	14	9	85	27	49	26	15	7	72	95
13/01/2014	13	10	97	26	21	62	57	72	96	65	51	19	80	63	17	48	63	18	44
14/01/2014	14	27	61	41	95	42	88	84	73	32	90	21	47	60	38	35	8	92	91
15/01/2014	15	65	67	69	31	94	51	83	15	83	87	26	31	45	51	61	54	79	52
16/01/2014	16	67	72	39	60	85	88	42	75	29	98	43	46	99	26	91	43	91	79
17/01/2014	17	71	34	52	37	18	65	91	82	27	87	15	0	16	11	2	84	81	55
18/01/2014	18	4	59	42	73	96	43	16	59	86	32	95	2	95	12	55	8	93	16
19/01/2014	19	17	22	79	75	71	61	66	79	87	94	35	53	25	69	40	44	36	49
20/01/2014	20	6	44	55	6	35	87	51	5	85	96	1	74	9	89	66	6	51	2
21/01/2014	21	77	30	13	67	19	26	64	67	86	65	79	35	26	17	21	3	96	14
22/01/2014	22	70	66	20	40	78	74	71	50	58	83	41	93	29	35	58	71	22	93

In the following example in Algeria on the day 4 (04/01/2014) the topic “war” is worth 55.

The software must start an analysis whenever it is executed. A user can run the script directly inside Rstudio compiler just clicking the *Run* button. The running time should be as low as possible.

If some of the following requests are not allowed by Twitter’s API or there is any other problem in making the software, please let me know.

Interface

Every change in the software must be performed modifying directly the script inside Rstudio. There is no need to create an interface or an executable version of the script. Clearly if it is possible all the user defined input variables should be present in the upper part of the code: In this way the user can find all the variables in the same place.

In order to connect to API, if you need to create a new Twitter profile please contact me first and then feel free to create a new user, submitting a gmail contact I will give to you.

Input

The software receives three types of variable:

- A *list of states* to analyze (in English)
- A *list of* words indicating the *topics* (in English)
- Two variables that define the lapse of time

The software receives a *list of states* as input. This list can be defined either as internal array or loaded from an external archive, as you prefer, but any user should be able to modify this list in any moment. For example if in the list there is a state that i don't want to analyze, I should be able to erase its name from the list without any problem.

The default list must contain all the Independent States in the World. You can find an example of this list in the website: http://www.state.gov/s/inr/rls/4250.htm

Another input is the *list of topics* to look for. This list can be defined either as internal array or loaded from an external archive, as you prefer, but any user should be able to modify this list in any moment. For example a user should be able to erase or add any word on the list and the simulation will search for posts that contain also that word.

The default list of words in the *list of topics* are:

- War
- Elections
- God
- Crisis
- Petroleum
- Sport
- Holiday

Moreover in the input variables there should be present two or more variables that refers to the lapse of time of the analysis. For example if I want to know the trend of these topics referred to a past period, I should be able to specify today's date and the number of days back in the past I want to analyze.

The analysis will took place in descending order, from past to present.

Input check

After having declared the variables, the software must check if there is any problem in their declaration. For example if I inserted in the list a state that doesn't exist or I declared a date in the future, the program must stop and an error message must appear.

Output

The output of the program must be an array that contains for each day how the single topics are worth in each state. The array must be stored both as internal variable and exported in a tsv (.txt) file.

Note that the numbers in the tables below are purely indicative.

Considering an example of only three topics, six countries and a starting date 01/01/2014, we would have an array like:

		Afghanistan			Albania			Algeria			Algeria			Argentina			Australia		
date	day	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections
01/01/2014	1	76	0	98	16	30	63	67	72	95	85	76	21	0	19	57	44	95	40
02/01/2014	2	15	4	54	61	49	46	46	76	98	25	32	76	1	45	46	12	93	40
03/01/2014	3	25	41	55	5	8	61	98	41	79	33	51	58	66	8	55	98	60	81
04/01/2014	4	51	74	71	67	82	79	55	33	37	68	61	70	88	63	90	15	33	80
05/01/2014	5	74	17	15	27	7	46	37	91	8	54	88	27	12	80	85	75	50	78
06/01/2014	6	90	27	2	79	43	80	62	41	41	74	91	89	60	58	97	73	92	71
07/01/2014	7	16	96	61	18	77	46	9	32	11	0	38	83	81	76	82	26	24	64
08/01/2014	8	13	10	3	17	28	3	23	17	63	90	93	36	40	2	81	37	66	16
09/01/2014	9	31	10	66	48	8	20	50	45	41	20	90	73	98	32	60	69	31	16
10/01/2014	10	52	29	82	86	61	49	62	28	4	38	88	90	12	58	6	44	26	89
11/01/2014	11	28	43	57	16	35	43	26	25	65	48	20	77	4	52	33	31	79	7
12/01/2014	12	63	67	41	38	50	52	77	99	14	9	85	27	49	26	15	7	72	95
13/01/2014	13	10	97	26	21	62	57	72	96	65	51	19	80	63	17	48	63	18	44
14/01/2014	14	27	61	41	95	42	88	84	73	32	90	21	47	60	38	35	8	92	91
15/01/2014	15	65	67	69	31	94	51	83	15	83	87	26	31	45	51	61	54	79	52
16/01/2014	16	67	72	39	60	85	88	42	75	29	98	43	46	99	26	91	43	91	79
17/01/2014	17	71	34	52	37	18	65	91	82	27	87	15	0	16	11	2	84	81	55
18/01/2014	18	4	59	42	73	96	43	16	59	86	32	95	2	95	12	55	8	93	16
19/01/2014	19	17	22	79	75	71	61	66	79	87	94	35	53	25	69	40	44	36	49
20/01/2014	20	6	44	55	6	35	87	51	5	85	96	1	74	9	89	66	6	51	2
21/01/2014	21	77	30	13	67	19	26	64	67	86	65	79	35	26	17	21	3	96	14
22/01/2014	22	70	66	20	40	78	74	71	50	58	83	41	93	29	35	58	71	22	93

From the picture above, the actual matrix (the one stored as internal output variable) would be:

01/01/2014	1	76	0	98	16	30	63	67	72	95	85	76	21	0	19	57	44	95	40
02/01/2014	2	15	4	54	61	49	46	46	76	98	25	32	76	1	45	46	12	93	40
03/01/2014	3	25	41	55	5	8	61	98	41	79	33	51	58	66	8	55	98	60	81
04/01/2014	4	51	74	71	67	82	79	55	33	37	68	61	70	88	63	90	15	33	80
05/01/2014	5	74	17	15	27	7	46	37	91	8	54	88	27	12	80	85	75	50	78
06/01/2014	6	90	27	2	79	43	80	62	41	41	74	91	89	60	58	97	73	92	71
07/01/2014	7	16	96	61	18	77	46	9	32	11	0	38	83	81	76	82	26	24	64
08/01/2014	8	13	10	3	17	28	3	23	17	63	90	93	36	40	2	81	37	66	16
09/01/2014	9	31	10	66	48	8	20	50	45	41	20	90	73	98	32	60	69	31	16
10/01/2014	10	52	29	82	86	61	49	62	28	4	38	88	90	12	58	6	44	26	89
11/01/2014	11	28	43	57	16	35	43	26	25	65	48	20	77	4	52	33	31	79	7
12/01/2014	12	63	67	41	38	50	52	77	99	14	9	85	27	49	26	15	7	72	95
13/01/2014	13	10	97	26	21	62	57	72	96	65	51	19	80	63	17	48	63	18	44
14/01/2014	14	27	61	41	95	42	88	84	73	32	90	21	47	60	38	35	8	92	91
15/01/2014	15	65	67	69	31	94	51	83	15	83	87	26	31	45	51	61	54	79	52
16/01/2014	16	67	72	39	60	85	88	42	75	29	98	43	46	99	26	91	43	91	79
17/01/2014	17	71	34	52	37	18	65	91	82	27	87	15	0	16	11	2	84	81	55
18/01/2014	18	4	59	42	73	96	43	16	59	86	32	95	2	95	12	55	8	93	16
19/01/2014	19	17	22	79	75	71	61	66	79	87	94	35	53	25	69	40	44	36	49
20/01/2014	20	6	44	55	6	35	87	51	5	85	96	1	74	9	89	66	6	51	2
21/01/2014	21	77	30	13	67	19	26	64	67	86	65	79	35	26	17	21	3	96	14
22/01/2014	22	70	66	20	40	78	74	71	50	58	83	41	93	29	35	58	71	22	93

while the names of the columns should be saved as column headers.

		Afghanistan			Albania			Algeria			Algeria			Argentina			Australia		
date	day	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections

Each row refers to a single day and hey are sorted in descending order. For example if we analyze 100 days in the past, the matrix will have 100 rows.

The first column contains the dates (the format is not so important), while the second one presents a progressive number that goes from 1 to the number of days analyzed. The date and numbers are in descending order, from the oldest to the most recent.

From the third one to the n^{th} column there must be the score of each topic repeated for each state. As you can see form the example below the topics “war”, “crisis” and “elections” are repeated although they refer to different states.

Afghanistan			Albania			Algeria			Algeria			Argentina			Australia		
war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections	war	crisis	elections

In this pattern the names of the states are in alphabetical order.

How it works

For each day in the specified interval of time and for each state present in the *list of states*, the software must count how many tweets contain the words specified in the *list of topics*.

In a given country, any post can be written either in native language or English, so the software must search words both in English (as specified in the list) and their translation to local language.
This translation must be fully automated. The user should define only the name of the state in the *list of states* and at the moment of the query the program should translate automatically.

How the scores are calculated

For example, consider that at the *day 2* in *Spain* we have a total of 1 million posts. The software will search among these posts all the topic words both in English and Spanish.

Only a small percentage of posts will include the words specified in the *list of topics*, say 100'000.

Suppose that the distribution for *day 2* in *Spain* is as follow:

War (Guerra)	15'000 posts	15.00%
Elections (Elecciones)	20'000 posts	20.00%
God (Dios)	2'000 posts	2.00%
Crisis (Crisi)	9'000 posts	9.00%
Petroleum (Gasolina)	14'000 posts	14.00%
Sport (Deporte)	30'000 posts	30.00%
Holiday (Vacaciones)	10'000 posts	10.00%
TOTAL	100'000 posts	100%

Over a total of 1M daily posts in Spain only 100K present at least one word present in the *list of topics*. The words counted are the ones present in the *list of topics* and their translation into Spanish. For example, in Spain, the posts that presents the word "war" and the ones that presents the word "Guerra" count for the same topic "war".

If a post contain more than a word of the *list of topics*, the post must count for both topics.

The numbers we are looking for, to be written in the output matrix, are the percentages indicated in the list (highlighted in blue). In this way we can say that at the *day 2* in *Spain*, the topic "war" is worth 15.00, while the "sport" is worth 30.00 and so on.

Note that given the fact that multiple words can be present inside a post, the sum of all percentages could differ form 100%. The decimal figures of the numbers must be two, as shown in the example above.

Personal solution

The following description is just a suggestion. I have absolutely no idea of how API works. Do whatever you want in order to obtain the output as indicated in the Output section.

The software starts the analysis the days from the past to present. For each day in the interval of dates the software must analyze what was posted on each state. The structure would present two nested *for cycles*: The main one that count the days and the nested one that analyze the states.

```
for(i in 1:nrow(days)){
        for(j in 1:nrow(states)){
                query(days[i],  states[j]) ….
        }
}
```

In this way it is much easier to compile the output file that presents in the first column the date in ascending order.