

# tnum\_assignment

2022-10-28

## True Numbers Assignment

One of the reasons for including Truenumbers and tnum in MA615 is that it gives you experience using R to work with software that approaches data and information in a new way. It also requires solving practical programming problems in R involving data in JSON and complex data frames. I have demonstrated some of this in class, but watching demonstrations does not teach you to code. Writing code does.

The Truenumbers assignment comes in two parts – one focused on text, the other on numeric analyses. I have described the text analysis portion of the assignment in class along with ways to approach the assignment using the tnum package along with the sentimentr package.

### Assignment Part 1: Text Analysis with Truenumbers

Use Truenumbers to produce a short book report about H.G. Wells’ *The Time Machine*. The Time Machine book report should be between two and four pages long and delivered as an HTML document generated with rmarkdown. Format the report with a title and your name at the top of the first page. Include at least one heading called “Conclusion”.

Use this assignment to explore True Numbers and your new found understanding of text analysis in R.

To get starte you’ll need to initialize True Numbers. If you still haven’t installed the package, run:

```
devtools::install_github("arazdow/tnum/tnum").
```

Next, fill in your credentials in the below code block and run it.

```
library(tnum) tnum.loadLibs() tnum.authorize(ip = "mssp.truenum.com:8080", creds = "yourEmail@bu.edu:yourPassword")
```

### Interacting with the Numberspace

Now that you are connected to the True Numbers server, you can use a handful of True Numbers functions to explore the number space. Namely `tnum.getAllProperties()`, `tnum.getAllSubjects()`, and `tnum.getAllTags()`. Each of these functions returns a list of strings about the specified portion of the available True Numbers. You might place the list in a data.frame or tibble, or even use some string methods to find points of interest. You will find the both *The Time Machine* and county-level demographic data for New England are already loaded in the MSSP number space.

### Queries

The most prevelant and useful function in the True Numbers package is `tnum.query()`. True Numbers query language uses a string composed of term prefixes. See the `tnumGettingStarted1.Rmd` for more information, the table of available terms has been included from that document for your reference anyway: This table covers the types of terms allowed in a query. Multiple terms must be separated by **and** or **or** boolean operators. These can be preceded by negation **not**, and can be grouped using parens.

prefix	term syntax	effect
subj:	path,#,*	matches subject
prop:	path,#,*	matches property
(no prefix)	path,#,*	matches tag
numval:	> 2.5, < 2.5, etc.	matches si numerical value
stringval:	regular expression	matches string or path values

```
## example ##
```

```
ex1 <- tnum.query("subj:New_Hampshire* and prop:population and numval:< 100000")
```

```
## Returned 0 results
```

## Tagging True Numbers

The function `tnum.addTag()` takes a True Numbers GUID, a string with the new tag, and a string with a brief note/description of the tag's purpose. To apply a tag to a set of true numbers from a query use `mapply`.

```
mapply(FUN = tnum.addTag,
       guid = ex1$id,
       tag = "county:small",
       text = "the smallest counties in New Hampshire")
```

```
## list()
```

```
# double check our tag was added successfully #
```

```
tnum.getTagsOfTn(ex1[1,]$id)
```

```
## NULL
```

```
# if your tag turns out to be incorrect, or you want to replace it
```

```
tnum.deleteTagFromNumberspace("county:small")
```

```
# deleteTagFromNumberspace removes the tag from EVERYWHERE in the numberspace
```

## Assignment part 2: Analysis of Numeric Data with Truenumbers

Using the above, make a statement about a set New England Counties. Consider what counties are wealthy or at risk, counties with large/small populations of higher education, etc. The statement you make should include ~10 true numbers. Complete the following steps: - Query the numberspace and tag numbers that fit your classification - Appropriately tag your observations, use the note to specify the feature your tag represents - Finally construct a kable table to display the numbers you tagged, their wrappers, values, and tags.