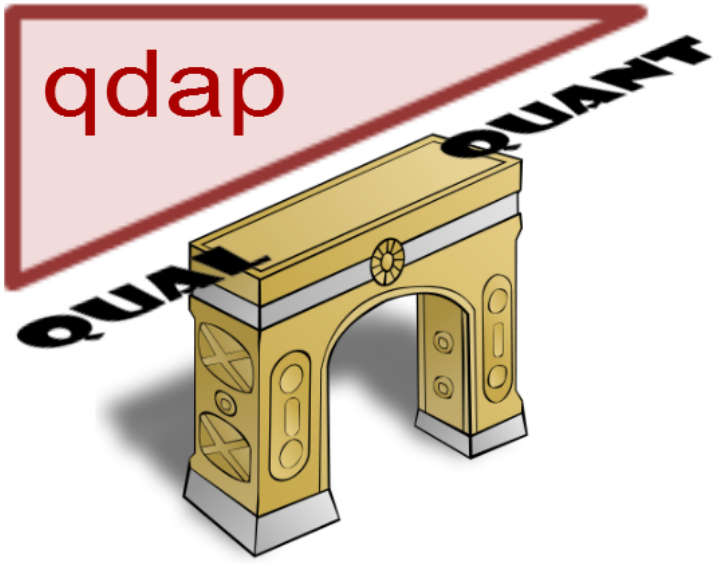
qdap-tm Package Compatability

Tyler W. Rinker

March 8, 2014



The **qdap** package (Rinker, 2013) is an R package designed to assist in quantitative discourse analysis. The package stands as a bridge between qualitative transcripts of dialogue and statistical analysis and visualization. The **tm** package (Feinerer and Hornik, 2014) is a major R (R Core Team, 2013) package used for a variety of text mining tasks. Many text analysis packages have been built around the **tm** package's infastructure (see CRAN Task View: Natural Language Processing). As **qdap** aims to act as a bridge to other R text mining analyses it is improtant that **qdap** provides a means of moving between the various **qdap** and **tm** data types. This vignette serves as a guide describign the various data formats of the two packages and the qdap functions that enable the user to move seemlessly between the two package.

1 Data Formats

The **qdap** and **tm** packages each have two basic data formats. **qdap** stores raw text data in the form of a `data.frame` augmented with columns of demogrphic variables whereas **tm** stores raw text as a `Corpus` and annotates demographic information with Meta Data attributes. The structures are both `lists` and are comparable.

The second format both packages use is a matrix structure of word frequency counts. The **qdap** package utilizes the *Word Frequency Matrix* (`wfm` function) whereas the **tm** package utilizes the *Term Document Matrix* or *Document Term Matrix* (`TermDocumentMatrix` and `DocumentTermMatrix` functions). Again the structure is similar between these two dataforms. Table 1 lays out the dataforms of the two packages.

Package	Raw Text	Word Counts
qdap	Dataframe	Word Frequency Matrix
tm	Corpus	Term Document Matrix/Document Term matrix

Table 1: **qdap-tm** Dataforms

One of the most visible differences between **qdap-tm** dataforms is that **qdap** enables the user to readily view the data while the **tm** utilizes a print method that provides a summary of the data. The `tm::inspect` function enables the user to view **tm** dataforms. The **qdap** package provides `qdap::qview` and `qdap::htruncdf` functions to view more digestable ammounts of the data. Let's have a look at the different data types. we'll start by loading both packages:

```r
library(qdap); library(tm)
```

Now let us have a look at the raw text storage of both packages.

1.1 Raw Text

1.1.1 qdap's Raw Text

```r
DATA
qview(DATA)
htruncdf(DATA)
```

```
## > DATA
##
##        person sex adult                                 state code
## 1         sam   m     0         Computer is fun. Not too fun.   K1
## 2        greg   m     0               No it's not, it's dumb.   K2
## .
## .
## .
## 9       sally   f     0           What are you talking about?   K9
## 10 researcher   f     1         Shall we move on?  Good then.  K10
## 11       greg   m     0 I'm hungry.  Let's eat.  You already?  K11


## > qview(DATA)
##
## ========================================================================
## nrow =  11           ncol =  5             DATA
## ========================================================================
##        person sex adult      state code
## 1         sam   m     0 Computer i   K1
## 2        greg   m     0 No it's no   K2
## .
## .
## .
## 8         sam   m     0 I distrust   K8
## 9       sally   f     0 What are y   K9
## 10 researcher   f     1 Shall we m  K10

## > htruncdf(DATA)
##
##        person sex adult      state code
## 1         sam   m     0 Computer i   K1
## 2        greg   m     0 No it's no   K2
## .
## .
## .
## 8         sam   m     0 I distrust   K8
## 9       sally   f     0 What are y   K9
## 10 researcher   f     1 Shall we m  K10
```

1.1.2 tm's Raw Text

```
data("crude")
crude
inspect(crude)
```

```
## > crude
## A corpus with 20 text documents
##
## > crude[[1]]
## Diamond Shamrock Corp said that
## effective today it had cut its contract prices for crude oil by
## 1.50 dlrs a barrel.
##     The reduction brings its posted price for West Texas
## Intermediate to 16.00 dlrs a barrel, the copany said.
##     "The price reduction today was made in the light of falling
## .
## .
## .
##     Diamond is the latest in a line of U.S. oil companies that
## have cut its contract, or posted, prices over the last two days
## citing weak oil markets.
##  Reuter
```

1.2 Word/Term Frequency Counts

Now we'll look at how the two packages handle word frequency counts. We'll start by setting up the raw text forms the two packages expect.

```
tm_dat <- qdap_dat <- DATA[1:4, c(1, 4) ]
rownames(tm_dat) <- paste("docs", 1:nrow(tm_dat))
tm_dat <- Corpus(DataframeSource(tm_dat[, 2, drop=FALSE]))
```

Both `qdap_dat` and `tm_dat` are storing this basic information:

```
qdap_dat

##    person                         state
## 1     sam Computer is fun. Not too fun.
## 2    greg       No it's not, it's dumb.
## 3 teacher            What should we do?
## 4     sam          You liar, it stinks!
```

1.2.1 qdap's Frequency Counts

```
with(qdap_dat, wfm(state, person))

##          greg researcher sally sam teacher
## computer    0          0     0   1       0
## do          0          0     0   0       1
## dumb        1          0     0   0       0
## fun         0          0     0   2       0
## is          0          0     0   1       0
## it          0          0     0   1       0
## it's        2          0     0   0       0
## liar        0          0     0   1       0
## no          1          0     0   0       0
## not         1          0     0   1       0
## should      0          0     0   0       1
## stinks      0          0     0   1       0
## too         0          0     0   1       0
## we          0          0     0   0       1
## what        0          0     0   0       1
## you         0          0     0   1       0
```

1.2.2 tm's Frequency Counts

Now we'll Look at the tm output using `inspect`.

```r
TermDocumentMatrix(tm_dat,
    control = list(
        removePunctuation = TRUE,
        wordLengths=c(0, Inf))
    )
)
```

```
## > TermDocumentMatrix(tm_dat)
## A term-document matrix (13 terms, 4 documents)
##
## Non-/sparse entries: 13/39
## Sparsity           : 75%
## Maximal term length: 8
## Weighting          : term frequency (tf)
```

```r
inspect(TermDocumentMatrix(tm_dat,
    control = list(
        removePunctuation = TRUE,
        wordLengths=c(0, Inf))
    )
))
```

```
#'           Docs
#' Terms      docs 1 docs 2 docs 3 docs 4
#'   computer      1      0      0      0
#'   do            0      0      1      0
#'   dumb          0      1      0      0
#'   fun           2      0      0      0
#'   is            1      0      0      0
#'   it            0      0      0      1
#'   its           0      2      0      0
#'   liar          0      0      0      1
#'   no            0      1      0      0
#'   not           1      1      0      0
#'   should        0      0      1      0
#'   stinks       0      0      0      1
#'   too          1      0      0      0
#'   we           0      0      1      0
#'   what         0      0      1      0
#'   you          0      0      0      1
```

The two matrices are esentially the same, with the execption of column order and names. Notice that we have to specify not to remove low character words while removing punctuation in the **tm** package semantics. qdap takes the opposite approach, removing punctuation by default and ustilizing all word lengths by default. These differences arise out of the intended uses, audiences, and philosophies of the package authors. Each has strengths in particular situations. The]textbfqdap output is an ordinary `matrix` whereas the **tm** output is a more compact `simple triplet matrix`. While the storage is different, both packages can be made to mimic the default of the other.

Also note that the **qdap** `sumamry` method for `wfm` provides the user with information simialr to the `TermDocumentMatrix`/`DocumentTermMatrix` functions' deault `print` method.

```
summary(with(qdap_dat, wfm(state, person)))

## A word-frequency matrix (16 terms, 5 groups)
##
##
## Non-/sparse entries      :  17/63
## Sparsity                  :  79%
## Maximal term length      :  8
## Less than four characters :  56%
## Hapax legomenon          :  13(81%)
## Dis legomenon            :  3(19%)
## Shannon's diversity index :  2.73
```

Now we'll look at some **qdap** functions that enable the user to move between packages, gaining the flexibility and benefits of both packages.

2 Converting Data Forms

We'll again use the following preset data.

```
tm_dat <- qdap_dat <- DATA[1:4, c (1, 4) ]
rownames (tm_dat) <- paste ("docs", 1: nrow (tm_dat))
tm_dat <- Corpus ( DataframeSource (tm_dat[, 2, drop=FALSE]))

qdap_wfm <- with (qdap_dat, wfm (state, person))
tm_tdm <- TermDocumentMatrix (tm_dat,
    control = list (
        removePunctuation = TRUE,
        wordLengths= c (0, Inf)
    )
)
```

1. `qdap_dat` – is a **qdap** raw text form
2. `tm_dat` – is a **tm** raw text format
3. `qdap_wfm` – is a **qdap** word frequncies count
4. `tm_tdm` – is a **tm** word frequncies count

2.1 Corpus to data.frame

To move from a `Corpus` to a `data.frame` the `tm_corpus2df` function is used as follows:

```
tm_corpus2df(tm_dat)

##     docs                          text
## 1 docs 1 Computer is fun. Not too fun.
## 2 docs 2       No it's not, it's dumb.
## 3 docs 3            What should we do?
## 4 docs 4          You liar, it stinks!
```

2.2 data.frame to Corpus

To move from a `data.frame` to a `Corpus` the `df2tm_corpus` function is used as follows:

```
with(qdap_dat, df2tm_corpus(state, person))

## A corpus with 3 text documents
```

2.3 TermDocumentMatrix/DocumentTermMatrix to wfm

To move from a TermDocumentMatrix to a wfm the as.wfm function is used as follows:

```
as.wfm(tm_tdm)

##          docs 1 docs 2 docs 3 docs 4
## computer      1      0      0      0
## do            0      0      1      0
## dumb          0      1      0      0
## fun           2      0      0      0
## is            1      0      0      0
## it            0      0      0      1
## its           0      2      0      0
## liar          0      0      0      1
## no            0      1      0      0
## not           1      1      0      0
## should        0      0      1      0
## stinks        0      0      0      1
## too           1      0      0      0
## we            0      0      1      0
## what          0      0      1      0
## you           0      0      0      1
```

2.4 wfm to TermDocumentMatrix/DocumentTermMatrix

To move from a wfm to a TermDocumentMatrix or DocumentTermMatrix the tdm and dtm functions can be used as follows:

```
tdm(qdap_wfm)

## A term-document matrix (16 terms, 5 documents)
##
## Non-/sparse entries: 17/63
## Sparsity           : 79%
## Maximal term length: 8
## Weighting          : term frequency (tf)

dtm(qdap_wfm)
```

```
## A document-term matrix (5 documents, 16 terms)
##
## Non-/sparse entries: 17/63
## Sparsity           : 79%
## Maximal term length: 8
## Weighting          : term frequency (tf)
```

References

Feinerer I, Hornik K (2014). *tm: Text Mining Package*. Version 0.5-10, URL `http://CRAN.R-project.org/package=tm`.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`.

Rinker TW (2013). *qdap: Quantitative Discourse Analysis Package*. University at Buffalo/SUNY, Buffalo, New York. Version 1.0.0, URL `http://github.com/trinker/qdap`.