quanteda

Quantitative Analysis of Textual Data

company
everyindustry
needs models competition
challenges however percent insights
ability better
provide organizations
trends example create now well
resources skills people services product
economy
store information
science public need global
states may use
used digital new techno
identify see also
free
education world analytics
report times
take
analytics potential growth sector
three business us
market social value health using
fraud large
citizens productivity research

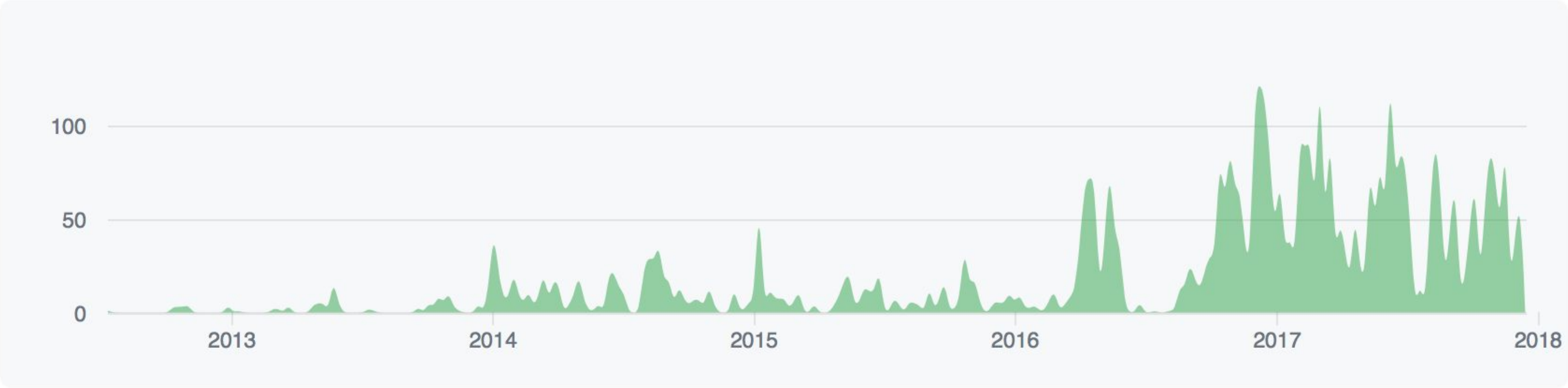
16 January 2018

Ken Benoit, Kohei Watanabe & Akitaka Matsuo
London School of Economics and Political Science

LSE SEDS
Social and Economic Data Science

Quantitative **An**alysis of **Te**xtual **Da**ta

- 5.5 years of development, 17 releases
- 6,791 commits; 719 issues; 8 core contributors
- > 93,000 downloads and rising



Contributions to master, excluding merge commits

The team

Ken Benoit (LSE)

Kohei Watanabe (LSE)

Aki Matsuo (LSE)

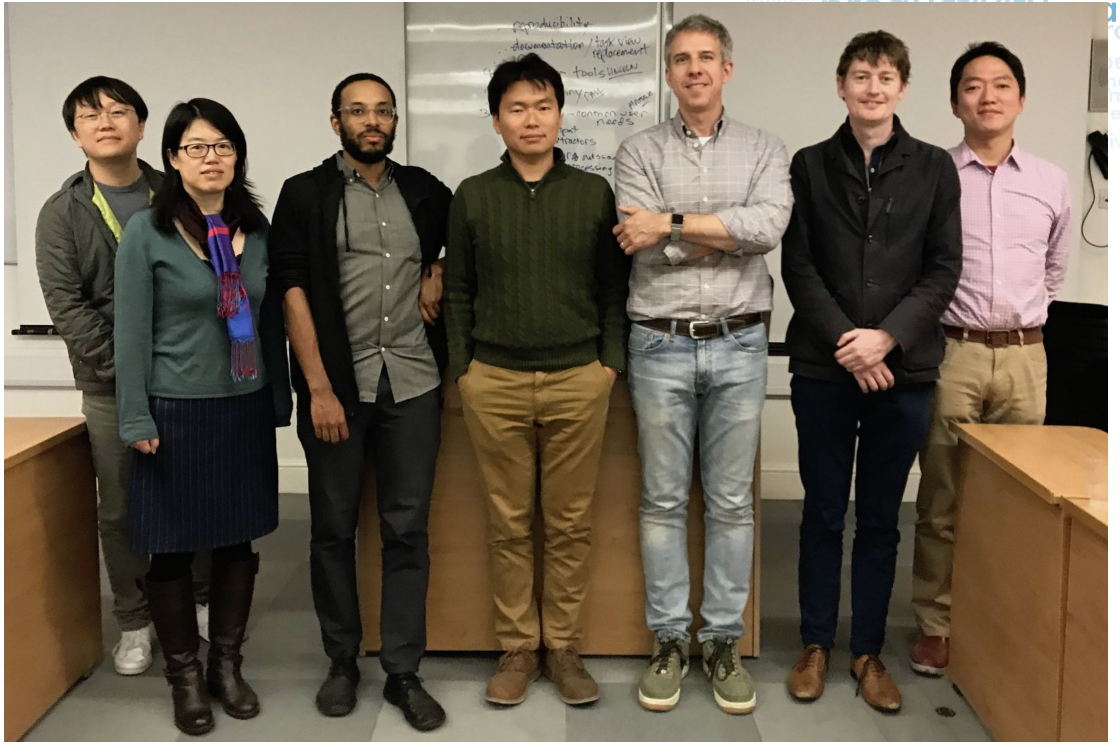
Haiyan Wang (De Beers)

Paul Nulty (Cambridge U)

Adam Obeng (Facebook)

Stefan Müller (Trinity College)

Ben Lauderdale (LSE)



Design of the package

- encourage analytic transparency and reproducibility
- have a consistent grammar - but use R idiom when natural, e.g. `summary()`
- be flexible enough for power users, simple enough for novices
- emphasize *performance*: use parallelization, hashing, and sparse matrices
- work nicely with other packages
- enable pipelined workflow using **magrittr**’s `%>%`

quanteda
Quantitative Analysis of Textual Data

corpus functions

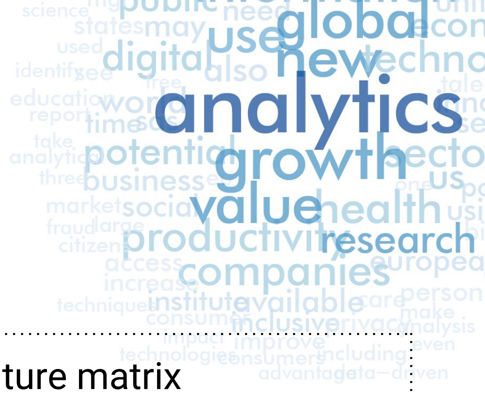
`corpus`	construct a corpus
`corpus_reshape`	recast the document units of a corpus
`corpus_sample`	randomly sample documents from a corpus
`corpus_segment`	segment texts into component elements
`corpus_subset`	extract a subset of a corpus
`corpus_trim`	remove sentences based on their token lengths or a pattern match

tokens functions

`tokens`	tokenize a set of texts
`tokens_compound`	convert token sequences into compound tokens
`tokens_lookup`	apply a dictionary to a tokens object
`tokens_select`, `tokens_remove`	select or remove tokens from a tokens object
`tokens_ngrams`, `tokens_skipgrams`	create ngrams and skipgrams from tokens
`tokens_tolower`, `tokens_toupper`	convert the case of tokens
`tokens_wordstem`	stem the terms in an object

`dfm`	create a document-feature matrix
`fcm`	create a feature co-occurrence matrix
`dfm_group`	recombine a dfm by grouping on a variable
`dfm_lookup`	apply a dictionary to a dfm
`dfm_sample`	randomly sample documents or features
`dfm_select, dfm_remove`	select features from a dfm or fcm
`dfm_sort`	sort a dfm by frequency of the margins
`dfm_tolower,dfm_toupper`	convert the case of the features of a dfm and combine
`dfm_trim`	trim a dfm using frequency threshold-based feature selection
`dfm_weight`	weight a dfm, including full SMART scheme, tf-idf, etc. in a dfm
`dfm_wordstem`	stem the features in a dfm

textmodel functions



`textmodel_ca`	correspondence analysis of a document-feature matrix
`textmodel_nb`	Naive Bayes (multinomial, Bernoulli) classifier for texts
`textmodel_wordfish`	Slapin and Proksch (2008) text scaling model
`textmodel_wordscores`	Laver, Benoit and Garry (2003) text scaling
`textmodel_affinity`	Perry and Benoit (2017) class affinity scaling

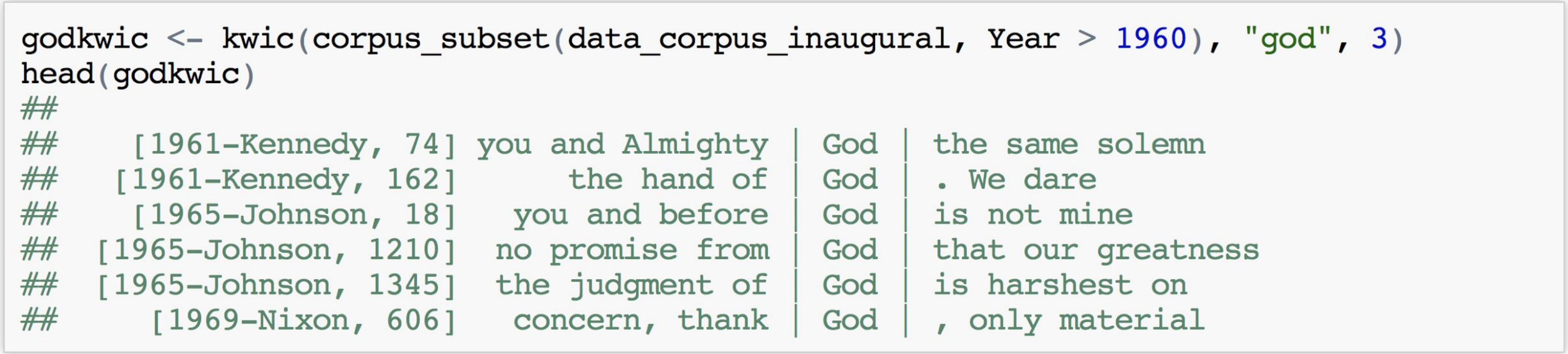
textstat functions

`textstat_collocations`	calculate collocation statistics
`textstat_dist`	distance computation between documents or features
`textstat_keyness`	calculate keyness statistics
`textstat_lexdiv`	calculate lexical diversity
`textstat_readability`	calculate readability
`textstat_simil`	similarity computation between documents or features

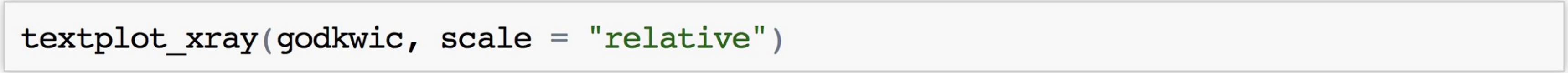
textplot functions

`textplot_scale1d`	plot a fitted scaling model
`textplot_wordcloud`	plot features as a wordcloud
`textplot_xray`	plot the dispersion of key word(s)
`textplot_keyness`	plot association of words with target v. reference set

Example: kwic()

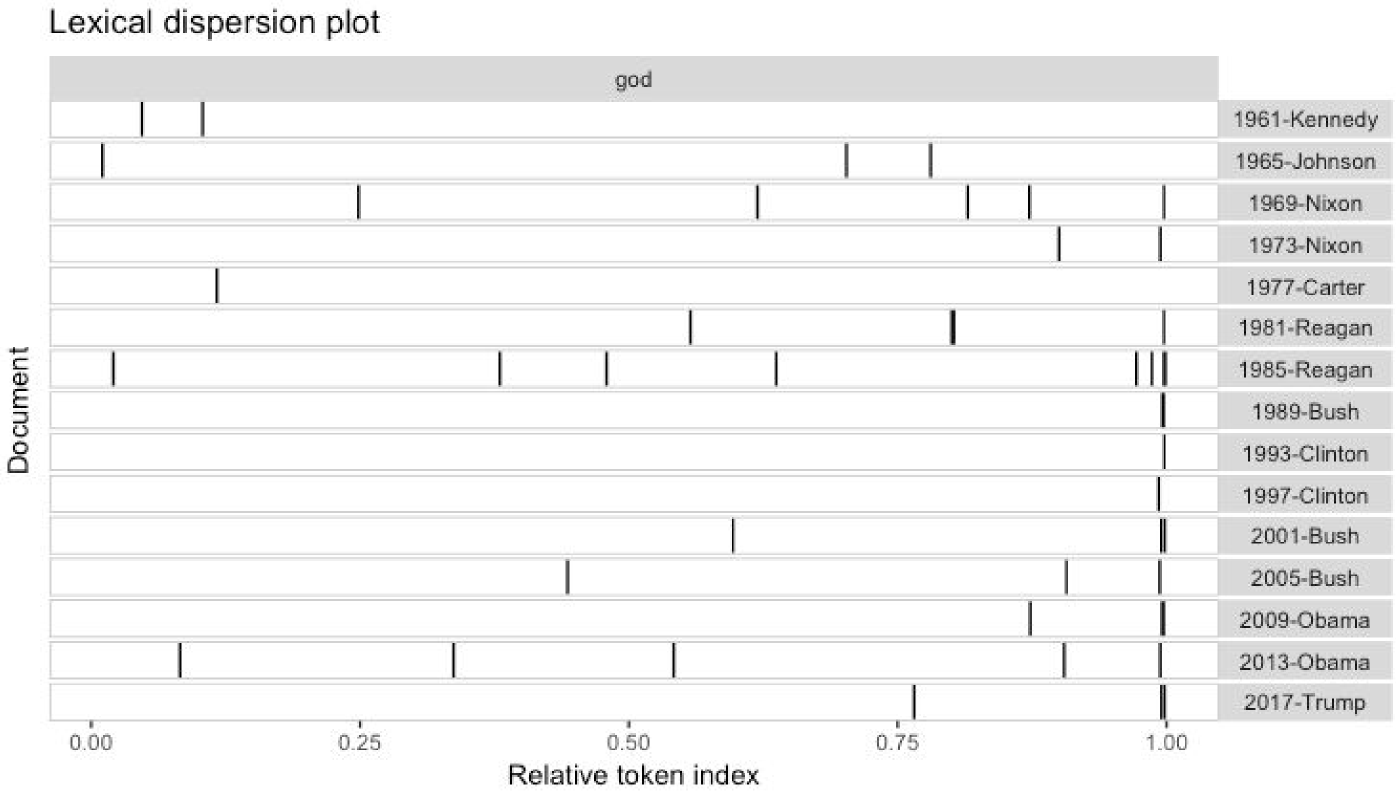


```
godkwic <- kwic(corpus_subset(data_corpus_inaugural, Year > 1960), "god", 3)
head(godkwic)
##
##    [1961-Kennedy, 74] you and Almighty | God | the same solemn
##   [1961-Kennedy, 162]        the hand of | God | . We dare
##    [1965-Johnson, 18]    you and before | God | is not mine
##  [1965-Johnson, 1210]   no promise from | God | that our greatness
##  [1965-Johnson, 1345]   the judgment of | God | is harshest on
##      [1969-Nixon, 606]   concern, thank | God | , only material
```



```
textplot_xray(godkwic, scale = "relative")
```

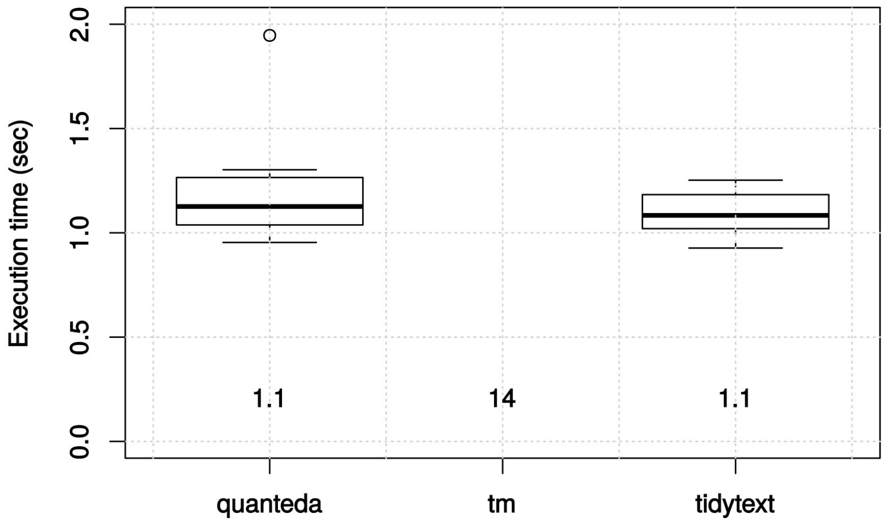
Example: `kwic()`



Performance

Tokenization

Tokenization using **stringi** to fully support Unicode



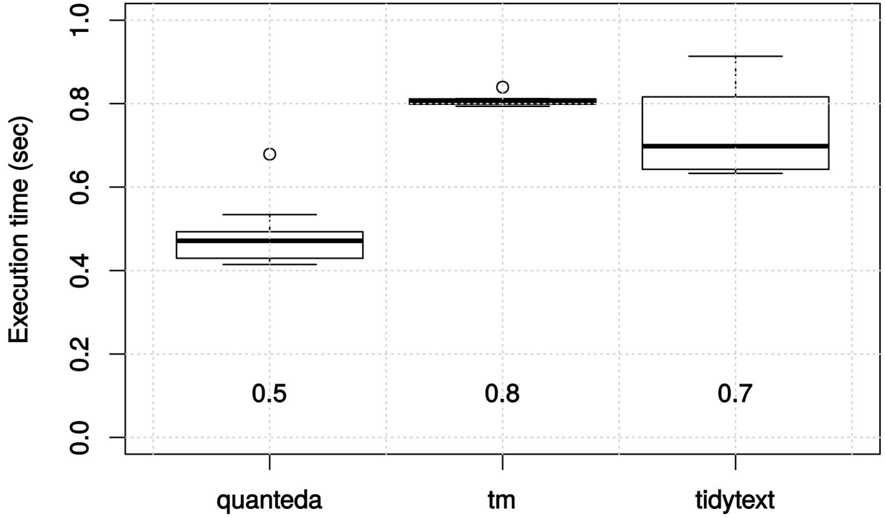
```
# quanteda commands

txt <- texts(data_corpus_guardian)
corp <- corpus(txt)

toks <- tokens(corp,
               what = "fastestword")
```

The gorpus contains 6,000 full-text Guardian news articles (10MB)

quanteda
Quantitative Analysis of Textual Data

Remove stopwords

Selection of tokens or sequences of tokens (multi-word expressions)



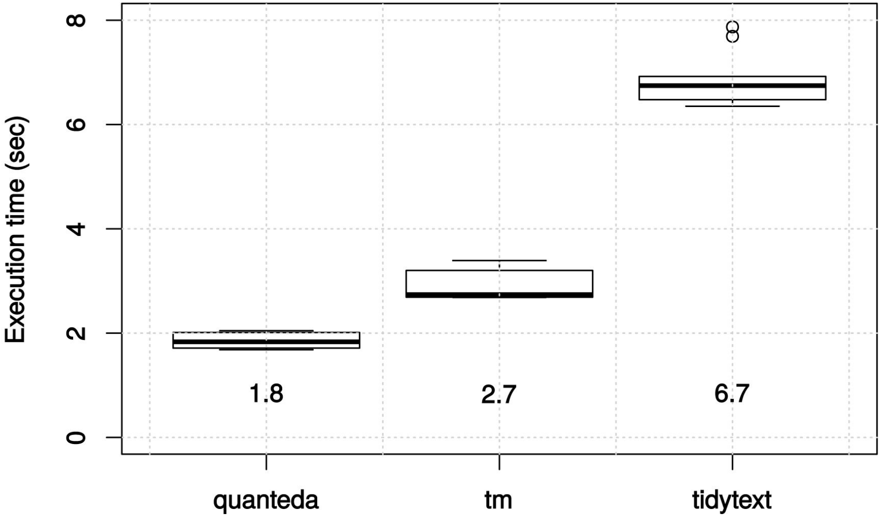
```
# quanteda commands

toks2 <- tokens_remove(toks, stopwords())
```

Stopwords contain 175 English function words from the stopwords() package

quanteda
Quantitative Analysis of Textual Data

Document-feature matrix

Tokenization and document-feature matrix construction using **Matrix**

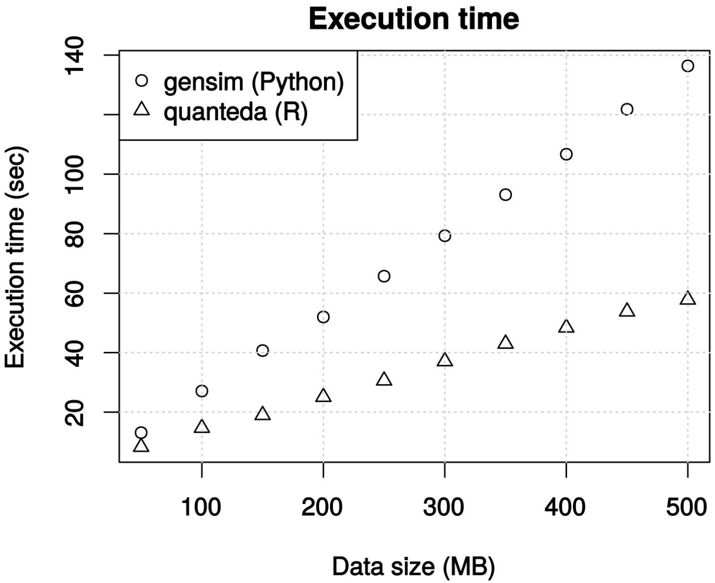


```
# quanteda commands

mt <- dfm(corp, what = "fastestword")
```

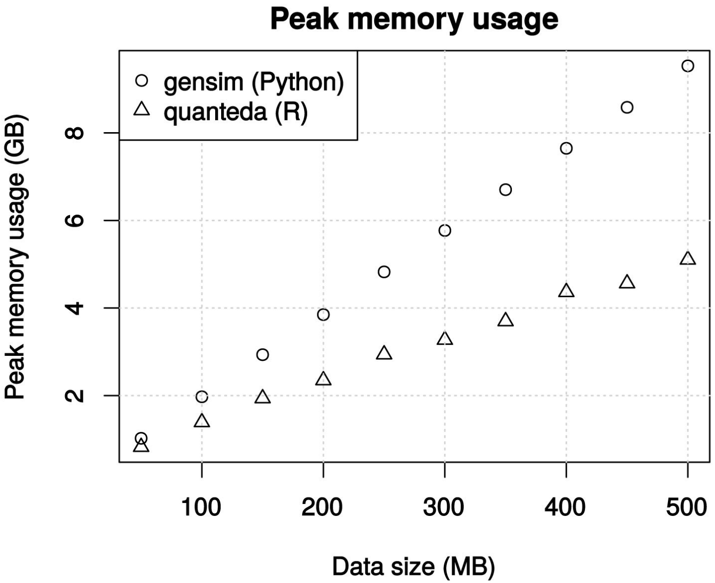
quanteda
Quantitative Analysis of Textual Data

Comparison with Python

2x more efficient than Python in speed and memory



Execution time



Peak memory usage

Test code is available at https://koheiw.net/?p=468

Secrets of high performance (1)

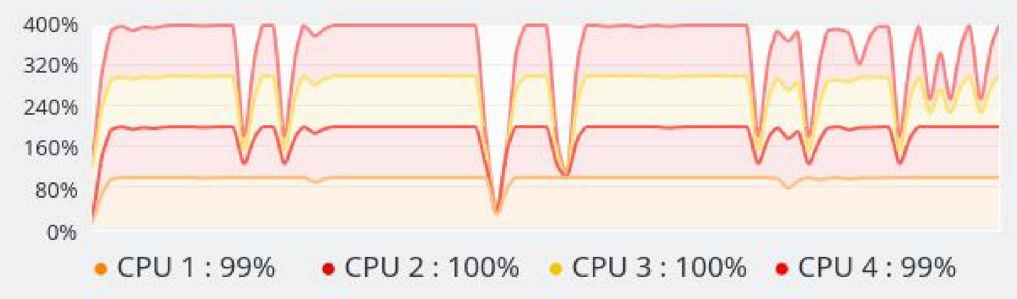
quanteda serializes tokens to speed up downstream operations

- Reduces RAM usage for tokens objects
  - Serialized tokens are 60-70% smaller than unserialized tokens
  - **quanteda** keeps tokens serialized from the beginning to the end
- Speeds up all the basic operations
  - Computers are faster with integers than characters
- Prevents Unicode characters to be garbled
  - Users can analyze Asian languages (e.g. Japanese and Chinese) using **quanteda**

Secrets of high performance (2)

quanteda implements parallel computing using **RcppParallel**

- Remove, lookup and compound operations on tokens are all parallelized in C++
  - Most effective in processing large number of documents
  - Allows complex rules for nuanced handling of sequences of tokens (multi-word expressions)
- Parallelization in C++ is much more efficient than in R
  - Shared memory parallelization has minimal overhead with large objects



quanteda
Quantitative Analysis of Textual Data

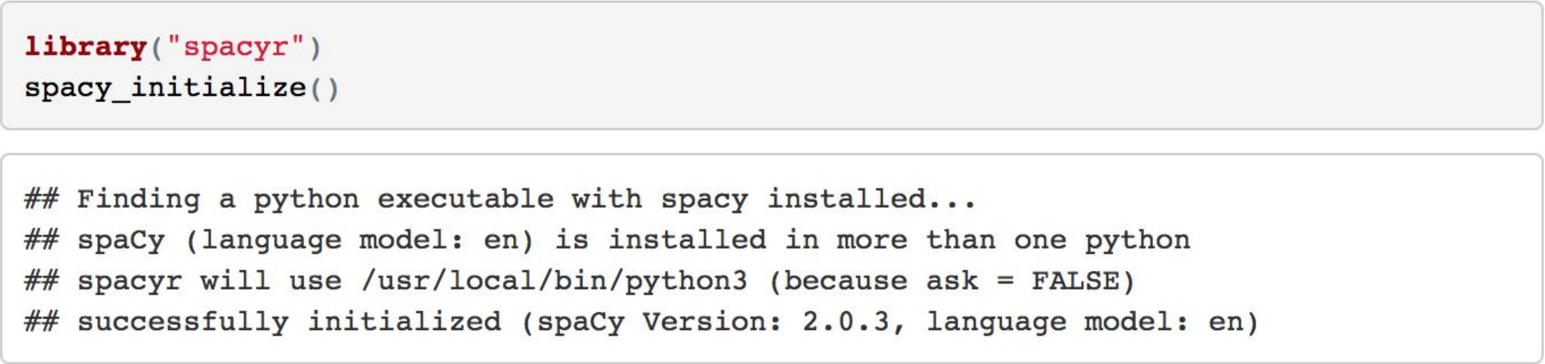
Accompanying packages

spacyr: an R wrapper for spaCy

spacyr is an R wrapper for spaCy (“Industrial-Strength Natural Language Processing” in Python)

- Returns data-frame of POS tagged tokens from text
- Options: POS-tagging, lemmatization, dependency parsing, named-entity extraction
- Using **reticulate** in backend
  - Solves most of cross platform compatibility issues
- Can use numerous language models in spaCy
  - e.g. English, German, French, Portuguese, Spanish
- Automatically detect spaCy installation from all python executables available in the system

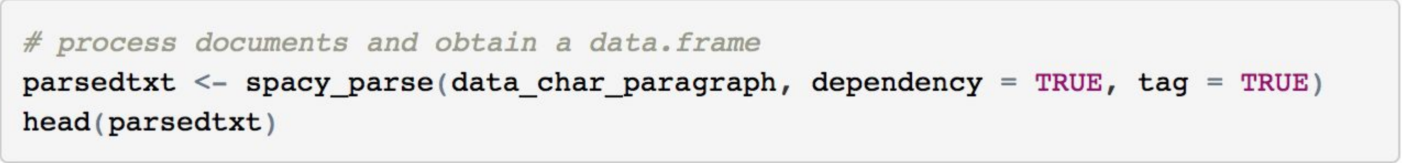
spacyr: initialize



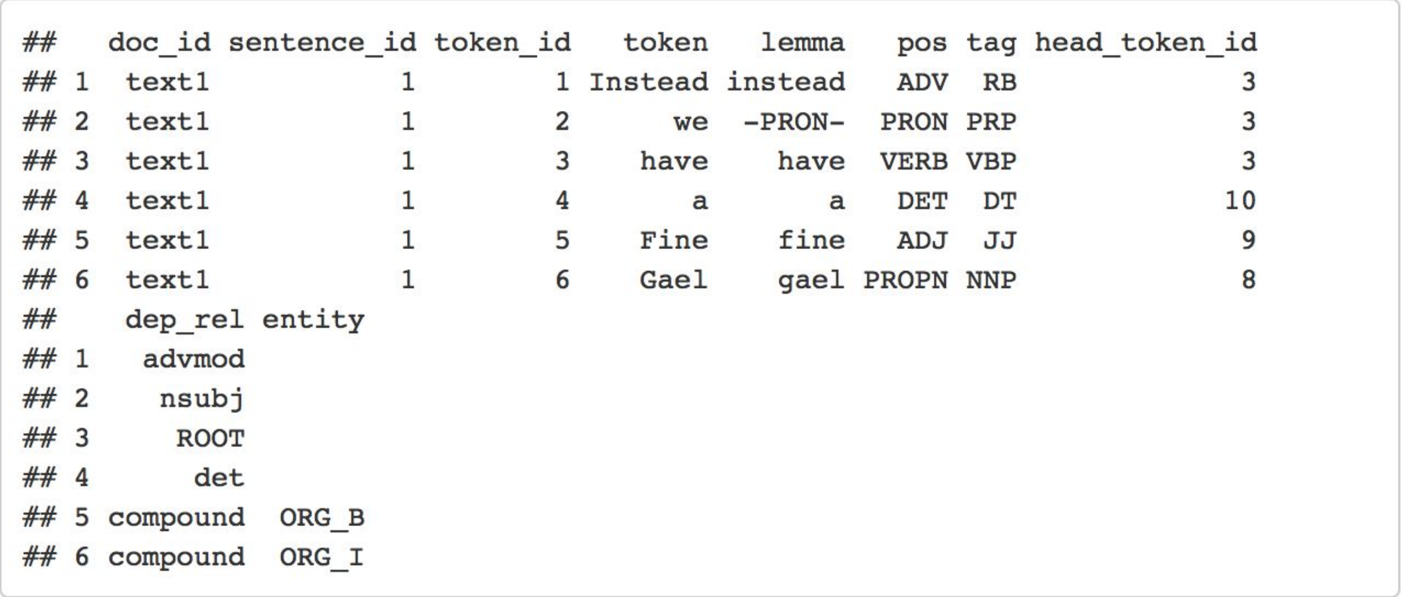
```r
library("spacyr")
spacy_initialize()
```

```
## Finding a python executable with spacy installed...
## spaCy (language model: en) is installed in more than one python
## spacyr will use /usr/local/bin/python3 (because ask = FALSE)
## successfully initialized (spaCy Version: 2.0.3, language model: en)
```

spacyr: basic parsing

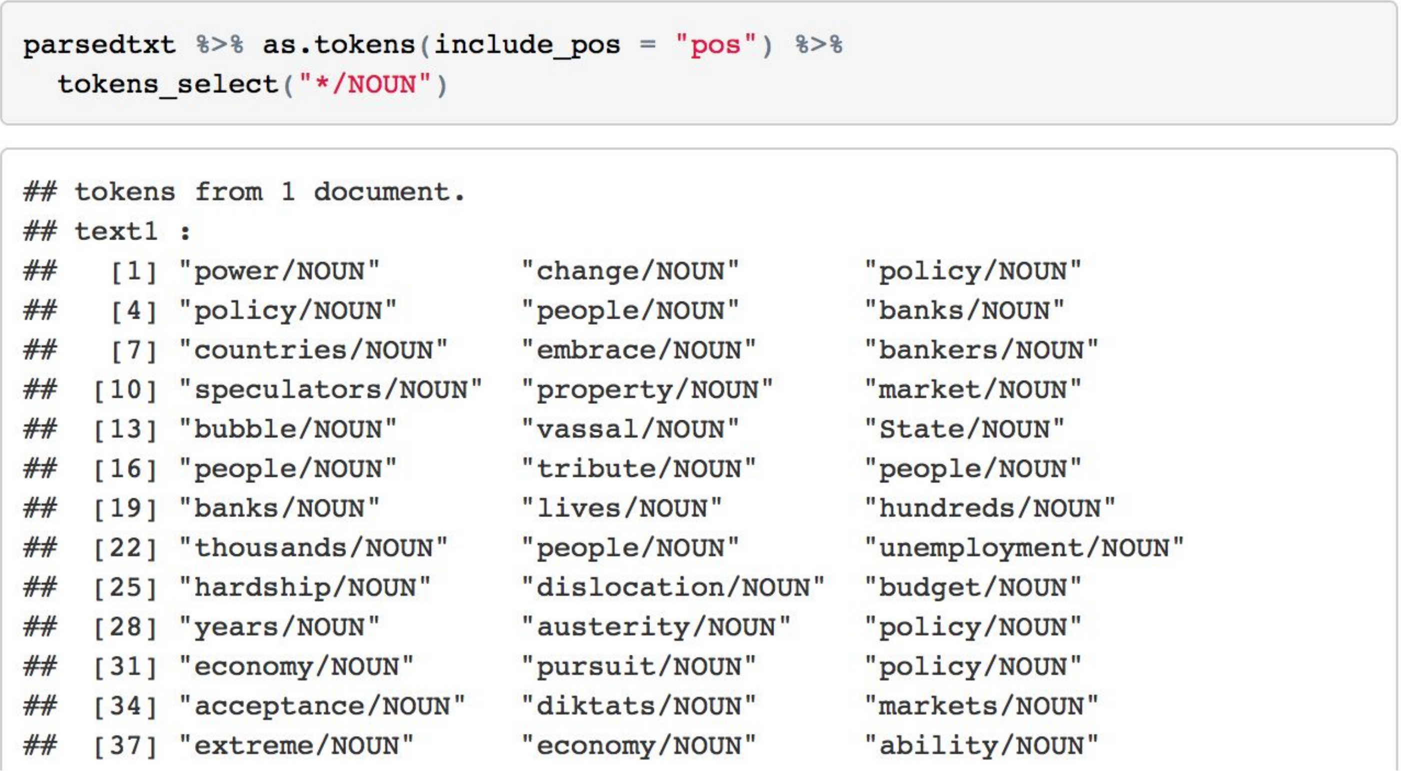


```
# process documents and obtain a data.frame
parsedtxt <- spacy_parse(data_char_paragraph, dependency = TRUE, tag = TRUE)
head(parsedtxt)
```



```
##    doc_id sentence_id token_id   token   lemma   pos tag head_token_id
## 1  text1           1        1 Instead instead   ADV  RB             3
## 2  text1           1        2      we  -PRON-  PRON PRP             3
## 3  text1           1        3    have    have  VERB VBP             3
## 4  text1           1        4       a       a   DET  DT            10
## 5  text1           1        5    Fine    fine   ADJ  JJ             9
## 6  text1           1        6    Gael    gael PROPN NNP             8
##     dep_rel entity
## 1    advmod
## 2     nsubj
## 3      ROOT
## 4       det
## 5 compound  ORG_B
## 6 compound  ORG_I
```

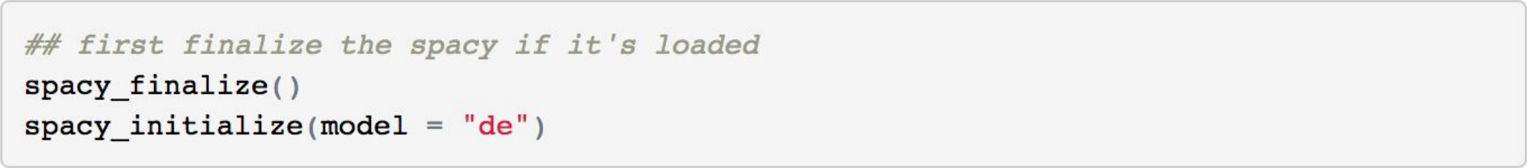
spacyr: connecting with quanteda



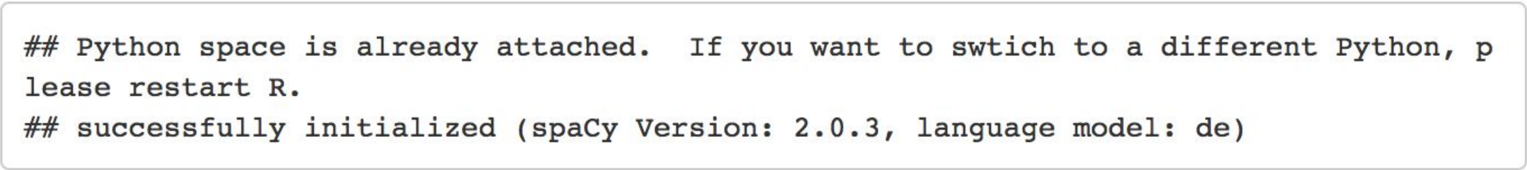
```
parsedtxt %>% as.tokens(include_pos = "pos") %>%
  tokens_select("*/NOUN")
```

```
## tokens from 1 document.
## text1 :
##   [1] "power/NOUN"       "change/NOUN"       "policy/NOUN"
##   [4] "policy/NOUN"      "people/NOUN"       "banks/NOUN"
##   [7] "countries/NOUN"   "embrace/NOUN"      "bankers/NOUN"
##  [10] "speculators/NOUN" "property/NOUN"     "market/NOUN"
##  [13] "bubble/NOUN"      "vassal/NOUN"       "State/NOUN"
##  [16] "people/NOUN"      "tribute/NOUN"      "people/NOUN"
##  [19] "banks/NOUN"       "lives/NOUN"        "hundreds/NOUN"
##  [22] "thousands/NOUN"   "people/NOUN"       "unemployment/NOUN"
##  [25] "hardship/NOUN"    "dislocation/NOUN"  "budget/NOUN"
##  [28] "years/NOUN"       "austerity/NOUN"    "policy/NOUN"
##  [31] "economy/NOUN"     "pursuit/NOUN"      "policy/NOUN"
##  [34] "acceptance/NOUN"  "diktats/NOUN"      "markets/NOUN"
##  [37] "extreme/NOUN"     "economy/NOUN"      "ability/NOUN"
```

spacyr: switching language models



```
## first finalize the spacy if it's loaded
spacy_finalize()
spacy_initialize(model = "de")
```



```
## Python space is already attached.  If you want to swtich to a different Python, p
lease restart R.
## successfully initialized (spaCy Version: 2.0.3, language model: de)
```

quanteda
Quantitative Analysis of Textual Data

spacyr: switching language models

```
txt_german <- c(R = "R ist eine freie Programmiersprache für statistische Berechnung
en und Grafiken. Sie wurde von Statistikern für Anwender mit statistischen Aufgaben
 entwickelt.",
               python = "Python ist eine universelle, üblicherweise interpretierte h
öhere Programmiersprache. Sie will einen gut lesbaren, knappen Programmierstil förde
rn.")
results_german <- spacy_parse(txt_german, dependency = TRUE, lemma = FALSE, tag = TR
UE)
head(results_german)
```

```
##   doc_id sentence_id token_id              token   pos   tag head_token_id
## 1      R           1        1                  R PROPN    NE             2
## 2      R           1        2                ist   AUX VAFIN             2
## 3      R           1        3               eine   DET   ART             5
## 4      R           1        4              freie   ADJ  ADJA             5
## 5      R           1        5 Programmiersprache  NOUN    NN             2
## 6      R           1        6                für   ADP  APPR             5
##   dep_rel entity
## 1      sb
## 2    ROOT
## 3      nk
## 4      nk
## 5      pd
## 6     mnr
```

readtext package

A one-function package that does exactly what it says on the tin:
It reads files containing text, along with any associated document-level metadata

- Available file formats: txt, csv, tsv, tab, json, xml, pdf, docx, doc, xls, xlsx, rtf
- Can multiple files at one time with
  - a wildcard value (filepath + glob)
  - url
  - file archives (e.g. tar, tar.gz, zip)

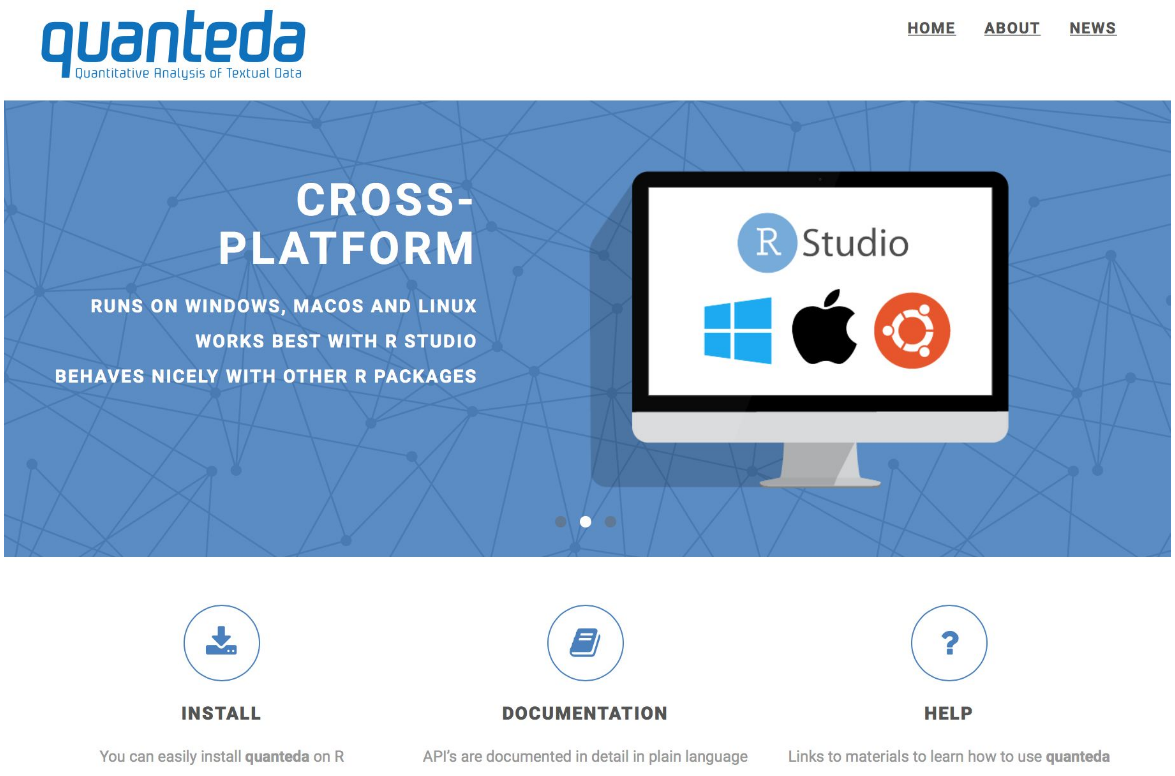
quanteda
Quantitative Analysis of Textual Data

Additional resources

quanteda

Quantitative Analysis of Textual Data

Portal site: *quanteda.io*



quanteda
Quantitative Analysis of Textual Data

CROSS-PLATFORM

RUNS ON WINDOWS, MACOS AND LINUX

WORKS BEST WITH R STUDIO

BEHAVES NICELY WITH OTHER R PACKAGES

INSTALL

You can easily install **quanteda** on R

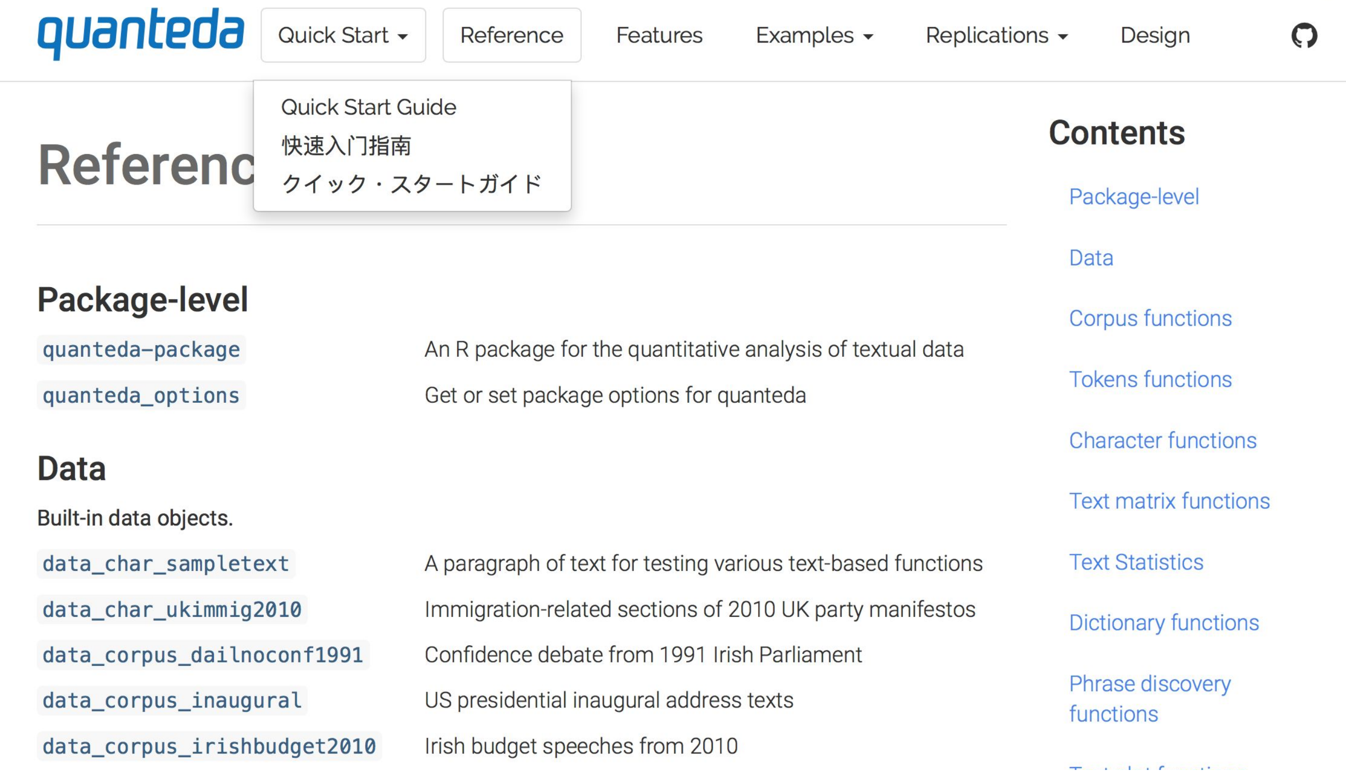
DOCUMENTATION

API's are documented in detail in plain language

HELP

Links to materials to learn how to use **quanteda**

Documentation: *docs.quanteda.io*



Referenc

Package-level

`quanteda-package`	An R package for the quantitative analysis of textual data
`quanteda_options`	Get or set package options for quanteda

Data

Built-in data objects.

`data_char_sampletext`	A paragraph of text for testing various text-based functions
`data_char_ukimmig2010`	Immigration-related sections of 2010 UK party manifestos
`data_corpus_dailnoconf1991`	Confidence debate from 1991 Irish Parliament
`data_corpus_inaugural`	US presidential inaugural address texts
`data_corpus_irishbudget2010`	Irish budget speeches from 2010

Contents

Official laptop stickers!

quanteda
Quantitative Analysis of Textual Data

もじ
كلمة
字
word
слово
quanteda
Quantitative Analysis of Textual Data
QI quanteda initiative

The future

- Big data performance
- (Better) Integration with external NLP libraries (e.g. spaCy)
- Integration with external machine learning libraries

QI quanteda initiative

QUANTEDA