





# Developers

Kenneth Benoit (LSE)

Kohei Watanabe (Waseda U)

Akitaka Matsuo (LSE)

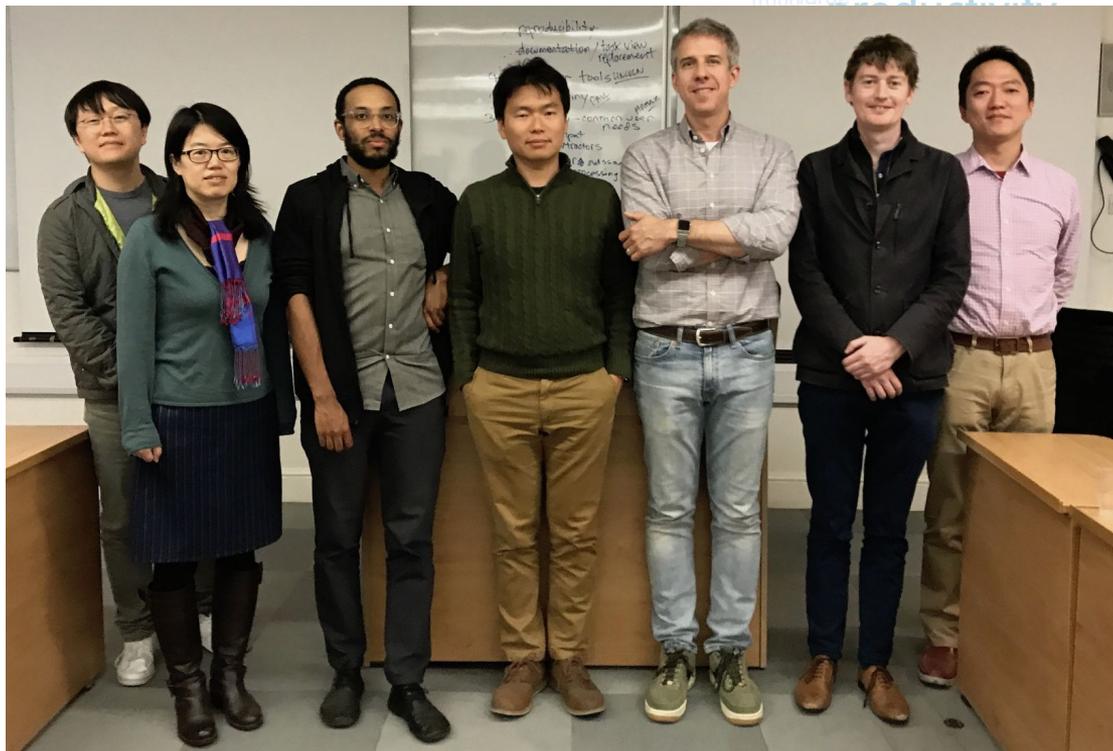
Haiyan Wang (De Beers)

Paul Nulty (Cambridge U)

Adam Obeng (Facebook)

Stefan Müller (Trinity College)

Benjamin Lauderdale (LSE)



# What is quantitative text analysis?

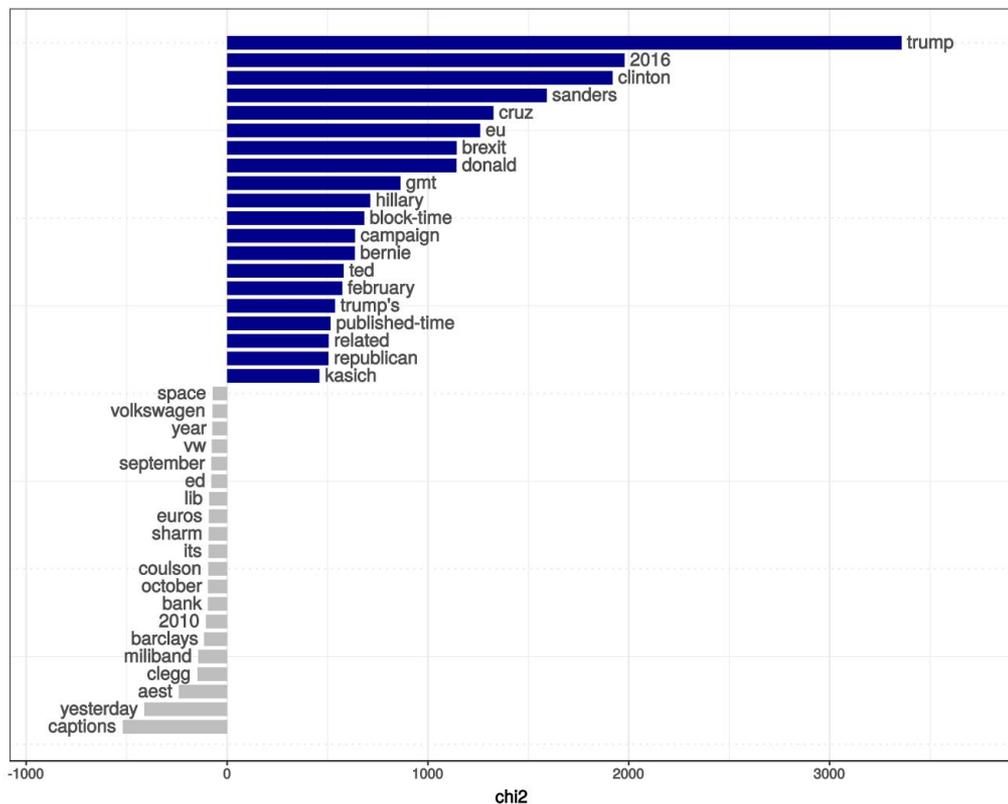
Application of NLP in social sciences and humanities in the context “big data”.

- Political Science
  - Estimate political ideology through election manifestos
  - Gauge complexity of political speeches
  - Monitor people’s response to election campaign on social media
- Media studies
  - Detect political bias in news articles
  - Extract mentions of political actors in new articles
- (Digital) humanity
  - Identify real authors of novels
  - Classify books according to the themes



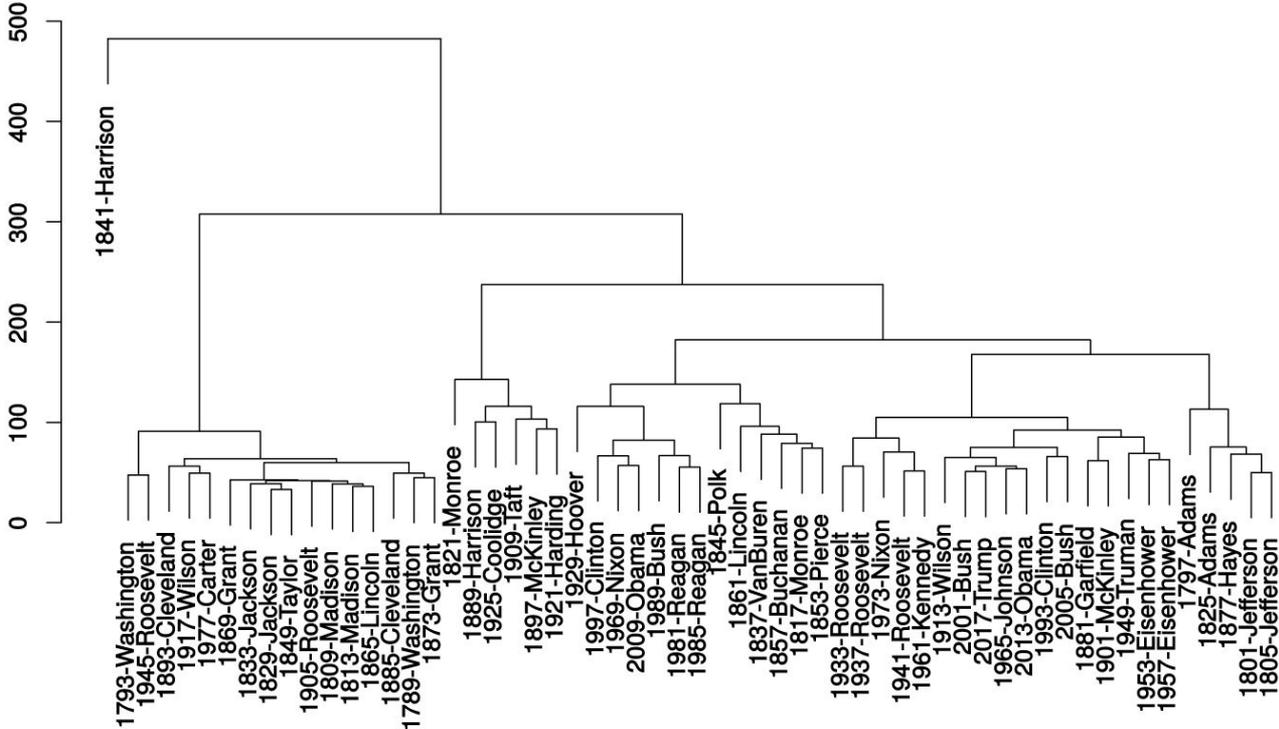


# Relative frequency (keyness)



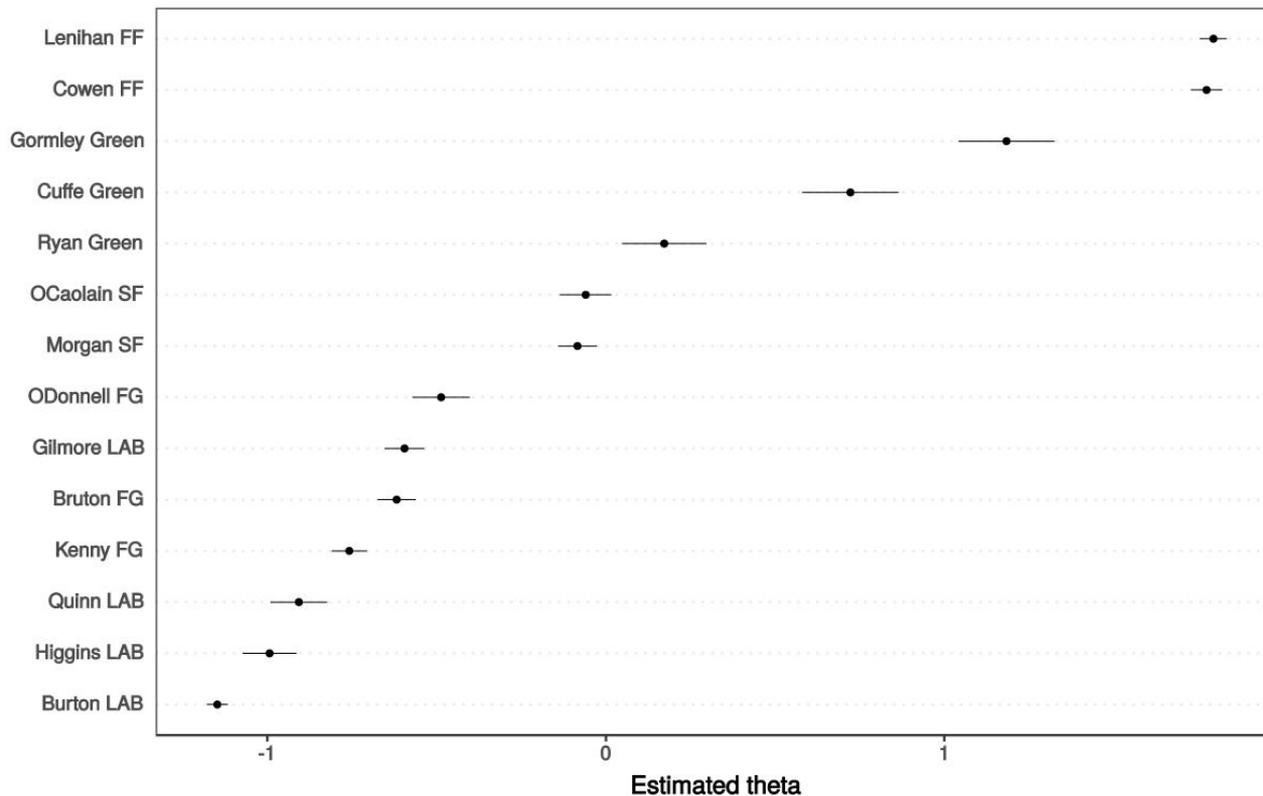
■ target  
■ reference

# Document similarity



public need global con  
states may new techno  
used digital also  
identific report  
education time  
take analytical  
three business  
market social value health usi  
fraud arge  
citizen  
address  
npanies  
available are person  
nclusiverivacn  
improve  
insumer including  
advantageta-driven

# Document scaling (wordfish)



science  
used  
identif  
education  
report  
take  
analytical  
three  
market  
fraud  
public  
states  
may  
need  
digital  
also  
new  
techno  
work  
times  
potential  
business  
social  
fraud  
global  
use  
digital  
also  
new  
techno  
growth  
value  
health  
research  
companies  
available  
inclusion  
improve  
consumer  
advantage  
data-driven







# Functions for corpus

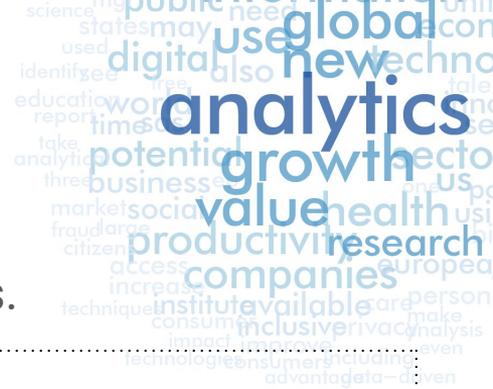
A **corpus** object contains texts with document-level variables.

<code>corpus()</code>	Construct a corpus
<code>corpus_reshape()</code>	Recast the document units of a corpus
<code>corpus_sample()</code>	Randomly sample documents from a corpus
<code>corpus_segment()</code>	Segment texts into component elements
<code>corpus_subset()</code>	Extract a subset of a corpus
<code>corpus_trim()</code>	Remove sentences based on their token lengths or a pattern match

# Functions for tokens

A **tokens** object contains individual words or symbols as tokens.

<code>tokens()</code>	Tokenize a set of texts
<code>tokens_compound()</code>	Convert token sequences into compound tokens
<code>tokens_lookup()</code>	Apply a dictionary to a tokens object
<code>tokens_select()</code> , <code>tokens_remove()</code>	Select or remove tokens from a tokens object
<code>tokens_ngrams()</code> , <code>tokens_skipgrams()</code>	Create ngrams and skipgrams from tokens
<code>tokens_tolower()</code> , <code>tokens_toupper()</code>	Convert the case of tokens
<code>tokens_wordstem()</code>	Stem the terms in an object



# Functions for document-feature matrix

A **dfm** object contains frequencies of words or symbols in a matrix.

<code>dfm()</code>	Create a document-feature matrix
<code>dfm_group()</code>	Recombine a dfm by grouping on a variable
<code>dfm_lookup()</code>	Apply a dictionary to a dfm
<code>dfm_select()</code> , <code>dfm_remove()</code>	Select features from a dfm or fcm
<code>dfm_trim()</code>	Trim a dfm using frequency threshold-based feature selection
<code>dfm_weight()</code>	Weight a dfm, including full SMART scheme, tf-idf, etc. in a dfm
<code>dfm_wordstem()</code>	Stem the features in a dfm
<code>fcm()</code>	Create a feature co-occurrence matrix

# Statistical analytic functions

**textstat\_\***() functions perform statistical analysis of textual data.

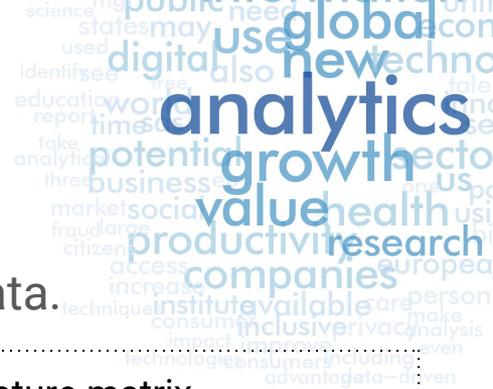
<code>textstat_collocations()</code>	Calculate collocation statistics
<code>textstat_dist()</code>	Distance computation between documents or features
<code>textstat_keyness()</code>	Calculate keyness statistics
<code>textstat_lexdiv()</code>	Calculate lexical diversity
<code>textstat_readability()</code>	Calculate readability
<code>textstat_simil()</code>	Similarity computation between documents or features



# Machine learning functions

**textmodel\_\*()** functions perform machine learning on textual data.

<code>textmodel_ca()</code>	Correspondence analysis of a document-feature matrix
<code>textmodel_lsa()</code>	Latent semantic analysis of a document-feature matrix
<code>textmodel_nb()</code>	Naive Bayes (multinomial, Bernoulli) classifier for texts
<code>textmodel_wordfish()</code>	Slapin and Proksch (2008) text scaling model
<code>textmodel_wordscores()</code>	Laver, Benoit and Garry (2003) text scaling
<code>textmodel_affinity()</code>	Perry and Benoit (2017) class affinity scaling
<code>covert()</code>	Interface to other R packages ( <b>topicmodels</b> , <b>stm</b> etc.)





# Asian-language support: Japanese

Japanese segmentation without morphological analysis tool (e.g. Mecab).

```
> txt_jp <- "政治とは社会に対して全体的な影響を及ぼし、社会で生きるひとりひとりの人の人生にも様々な影響を及ぼす複雑な領域である。"  
> quanteda::tokens(txt_jp)  
tokens from 1 document.  
text1 :  
 [1] "政治"      "と"      "は"      "社会"    "に対して"  
 [6] "全体"      "的"      "な"      "影響"    "を"  
[11] "及"        "ぼ"      "し"      "、"      "社会"  
[16] "で"        "生きる" "ひとりひとり" "の"      "人"  
[21] "の"        "人生"    "に"      "も"      "様々"  
[26] "な"        "影響"    "を"      "及ぼす" "複雑"  
[31] "な"        "領域"    "で"      "ある"    "。"
```

# Asian-language support: Chinese

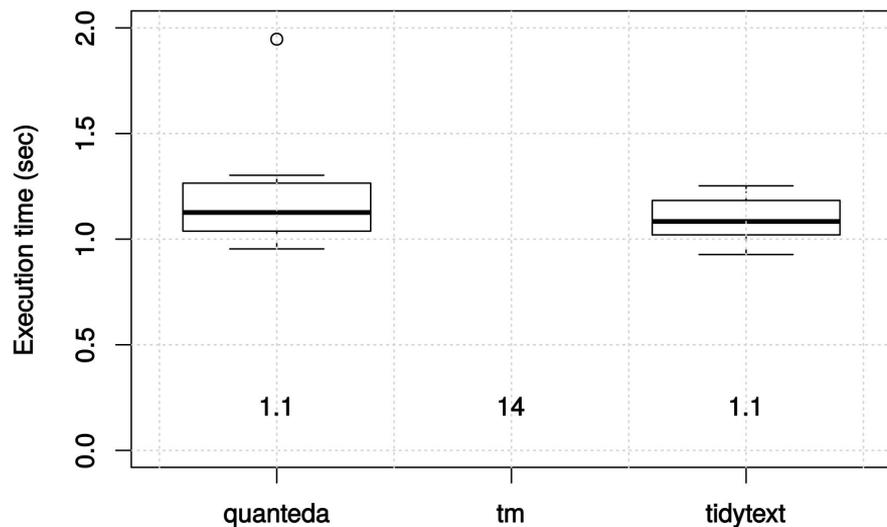
Chinese segmentation without morphological analysis (e.g. jieba)

```
> txt_cn <- "政治是各种團體进行集体决策的一个过程，也是各种團體或个人为了各自的領域所结
成的特定关系，尤指對於某一政治實體的統治，例如統治一個國家，亦指對於一國內外事務之監督
與管制。"
> quanteda::tokens(txt_cn)
tokens from 1 document.
text1 :
[1] "政治" "是" "各种" "团" "體" "进行" "集体" "决策"
[9] "的" "一个" "过程" ", " "也是" "各种" "团" "體"
[17] "或" "个人" "为了" "各自" "的" "領域" "所" "结成"
[25] "的" "特定" "关系" ", " "尤" "指" "對於" "某一"
[33] "政治" "實體" "的" "統治" ", " "例如" "統治" "一個"
[41] "國家" ", " "亦" "指" "對於" "一" "國內外" "事務"
[49] "之" "監督" "與" "管制" "。"
```



# Tokenization

Tokenization using **stringi** to fully support Unicode



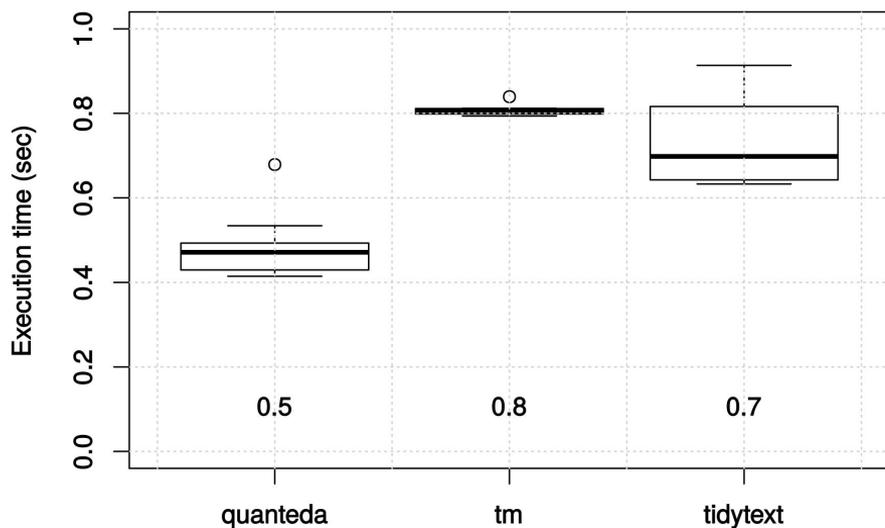
```
# quanteda commands
```

```
txt <- readLines("guardian.txt")  
corp <- corpus(txt)  
toks <- tokens(corp,  
               what = "fastestword")
```

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

# Remove stopwords

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



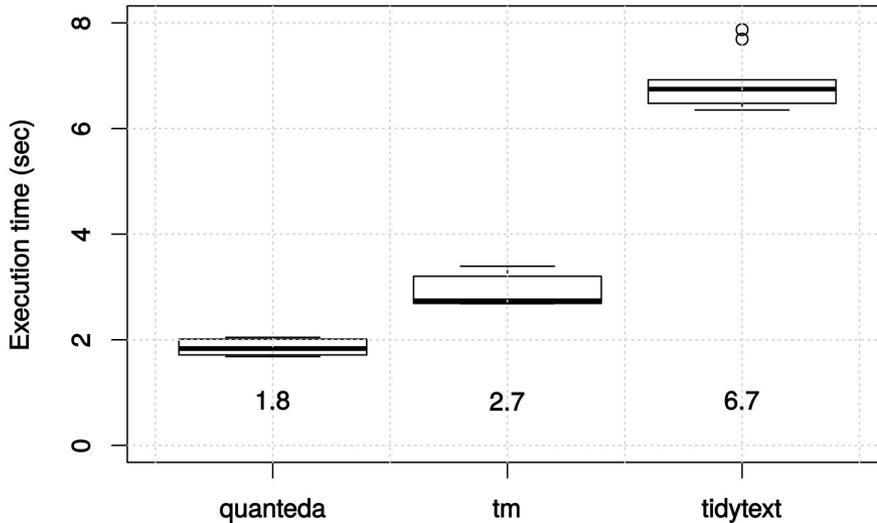
```
# quanteda commands
```

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

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

# Document-feature matrix

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



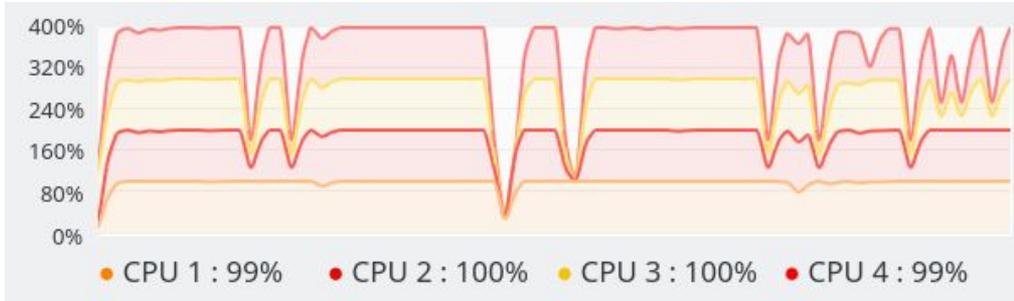
```
# quanteda commands
```

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



# Secrets of high performance

- Data serialization
  - Tokens are serialized to reduce memory usage by 50 to 70%
  - Speeds up all the basic operations
- Parallel computation
  - Many of the `tokens_*`() and `textstat_*`() functions are parallelized using **RcppParallel**
  - Parallelization in C++ is much more efficient than in R



science states may use global con  
used digital also new techno  
identific report times  
education worke  
take analytici  
three business  
market social value health usi  
fraud large citizen access companies european  
increase institut available care person  
consumers impact improve inclusiverivacy  
technologies consumers including even  
advantage data-driven



# Documentation website

<https://docs.quanteda.io>

**quanteda** 1.3.0 Quick Start ▾ Reference Features ▾ Examples ▾ Replications ▾ Search...

**quanteda: Quantitative Analysis of Textual Data**

**About**

An R package for managing and analyzing text, created by [Kenneth Benoit](#). Supported by the European Research Council grant ERC-2011-STG 283794-QUANTESS.

**How to Install**

The normal way from CRAN, using your R GUI or

```
install.packages("quanteda")
```

Or for the latest development version:

```
# devtools package required to install quanteda from Github
devtools::install_github("quanteda/quanteda")
```

Because this compiles some C++ and Fortran source code, you will need to have installed the appropriate compilers.

**If you are using a Windows platform**, this means you will need also to install the [Rtools](#) software available from CRAN.

**If you are using macOS**, you should install the [macOS tools](#), namely the Clang 6.x compiler and the GNU Fortran compiler (as [quanteda](#) requires gfortran to build).

**Links**

Download from CRAN at <https://cloud.r-project.org/package=quanteda>

Report a bug at <https://github.com/quanteda/quanteda/issues>

**License**

GPL-3

**Citation**

[Citing quanteda](#)

**Developers**

Kenneth Benoit  
Maintainer, author, copyright holder

Kohei Watanabe  
Author

Haiyan Wang  
Author

Paul Nulty  
Author

Adam Obeng  
Author

Stefan Müller  
Author

Akitaka Matsuo

**quanteda**  
Quantitative Analysis of Textual Data

science, states, may, need, global, con, used, digital, use, also, new, techno, identify, see, report, education, work, analytics, time, take, potential, growth, us, business, value, health, market, social, fraud, large, citizen, access, companies, european, increase, institute, available, care, person, consumer, impact, improve, make, technology, consumer, including, even, advantage, data-driven

# Tutorials website and workshops

<https://tutorials.quanteda.io>

**quanteda**  
Quantitative Analysis of Textual Data

Search

- 1. Introduction ✓
- 2. Data import
- 3. Basic operations
- 4. Statistical analysis
- 5. Advanced operations
- 6. Scaling and classification ✓
- 7. Different languages

MORE

- GitHub repo
- Website
- Clear History

## QUANTEDA TUTORIALS

By Kohei Watanabe and Stefan Müller

This website contains a step-by-step introduction to quantitative text analysis using **quanteda**. The chapters cover a brief introduction to the statistical programming language R, how to import text data, basic operations of **quanteda**, how to construct a corpus, tokens objects, a document-feature matrix, and how to conduct advanced operations. The final chapter deals with text scaling (e.g., Wordscores, Wordfish, correspondence analysis), document classification using Naive Bayes and topic models.

The six chapters consist of over 30 sections. If you click on the name of a chapter on the left-hand side of this page, the sections will pop up. You can also use the "Search" field in the top-left corner to look up the occurrence of certain terms or R functions covered in the tutorials.

This website is created for workshops held by the **quanteda** team and for users who look for a comprehensible step-by-step introduction to text analysis using R. We have also created several additional useful **resources**, such as vignettes, replications, a cheatsheet and a comparison to other text analysis packages (in terms of **functions** and **performance**) to get you started.

You can not only see the R commands but execute them yourself if you **download the source code of this website** from the **GitHub repository**. You should unzip the files on your machine and click **quanteda\_tutorials.Rproj** to open RStudio. Executable R commands are in the **.Rmarkdown** files under the **content** folder.

Contributions in the form of feedback, comments, code, and **bug** reports are most welcome. If you have questions on how to use **quanteda**, please post them to **the quanteda channel on StackOverflow**, if you find a bug, please report it to the **quanteda issues**. *We prefer these platforms to emails in communicating with our users* because the records will help other users who have similar problems.

**Info**

Examples in this tutorial are written for **quanteda** version 1.3.0. Please check if you have the same version installed by a command `packageVersion('quanteda')`.

**quanteda**  
Quantitative Analysis of Textual Data

