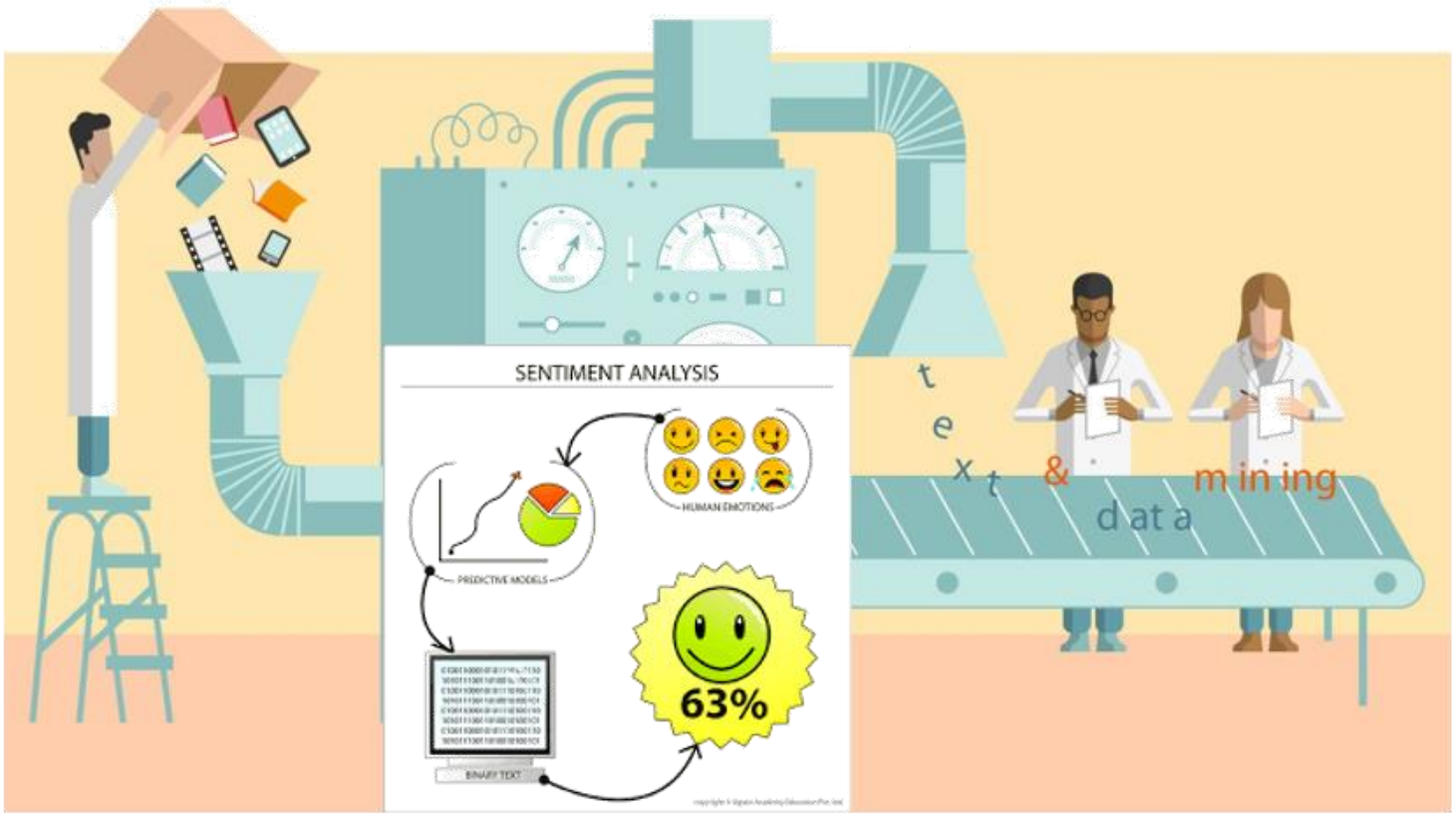
The R Package sentometrics to Compute, Aggregate and Predict with Textual Sentiment

David Ardia
Keven Bluteau
Samuel Borms (Université de Neuchâtel/Vrije Universiteit Brussel)
Kris Boudt

R/Finance 2018

Text mining…

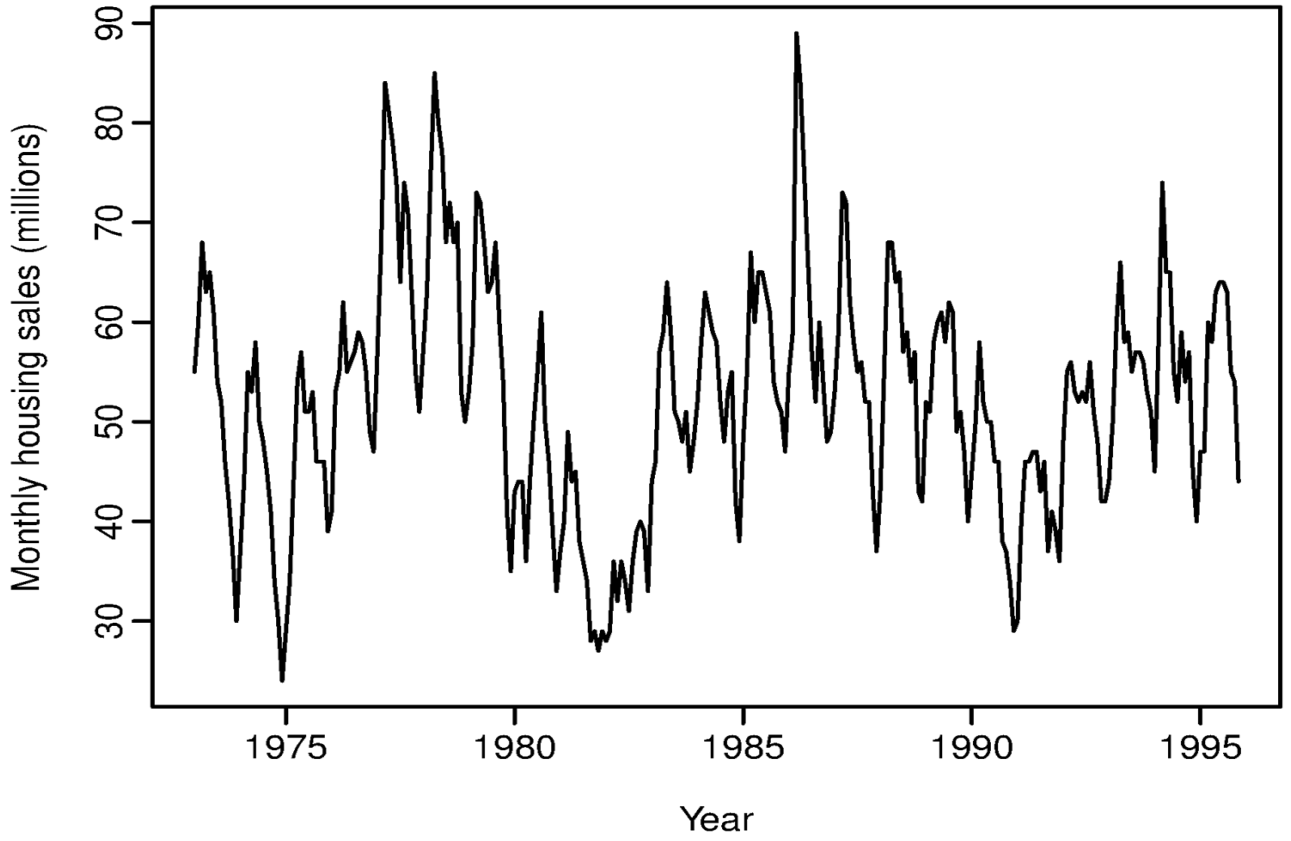
… is the process of distilling actionable insights from text.



Our focus is on textual sentiment analysis.

Time series econometrics…

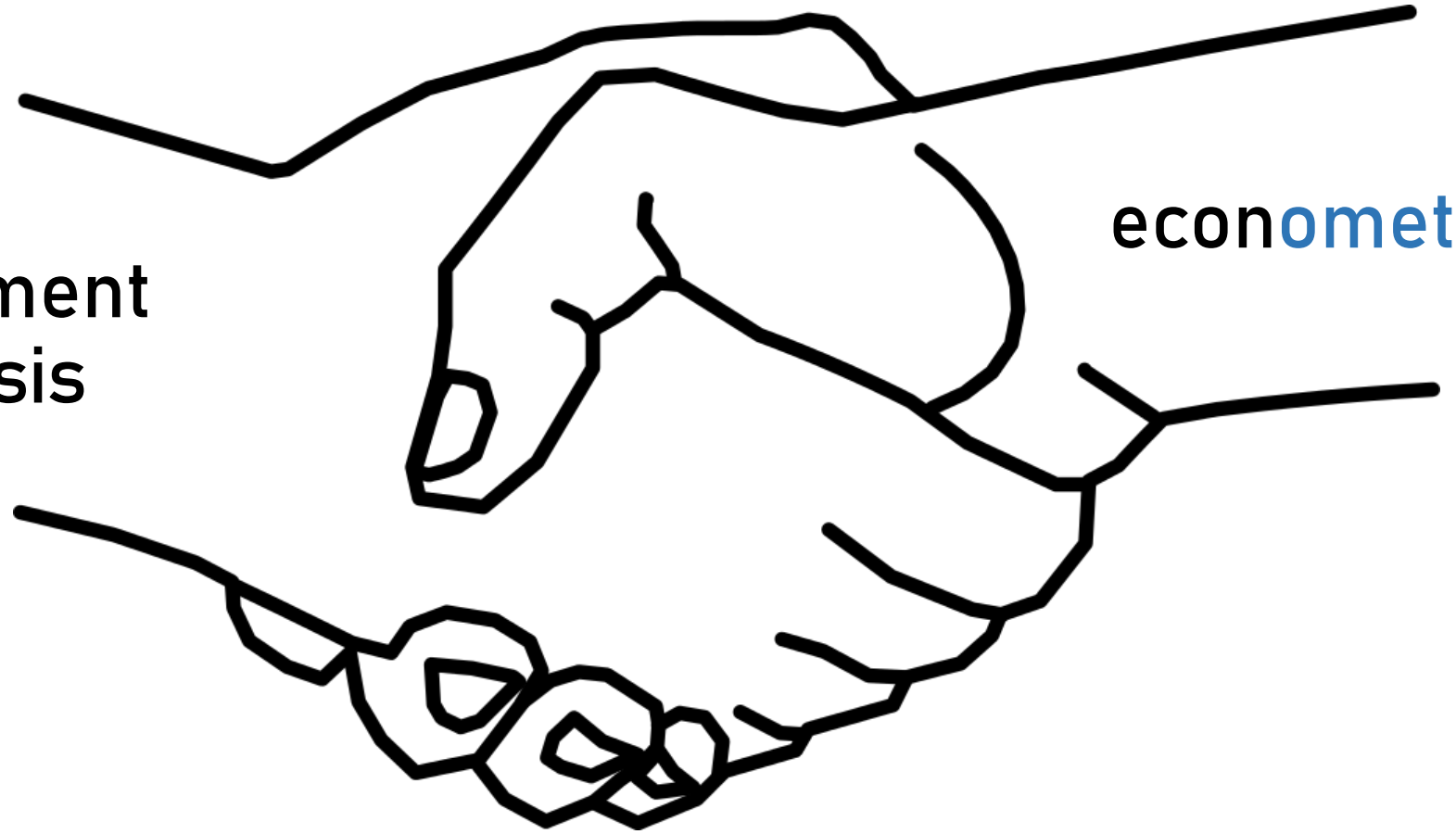
… is the analysis of quantitative time series data typically in an economic context.



Our focus is on aggregation, econometric modelling and prediction.

sentiment
analysis

econometrics



sentometrics

research ⟷ R package

The R package sentometrics

The package is a general framework that integrates (i) the qualification of sentiment from texts, (ii) the aggregation into different sentiment measures, and (iii) the optimized prediction based on these measures.

Typical workflow:

STEP 1
Build a corpus of texts with quantifiable metadata (“features”)

STEP 2
Pick lexicons and compute textual sentiment

STEP 3
Aggregate document-level sentiment scores into time series

STEP 4
Estimate a sentiment-based prediction model

STEP 5
Evaluate model performance and sentiment attribution

Let's go for a run with sentometrics

```
library("sentometrics")
```

We have a built-in dataset of news articles between 1995 and 2014, from The Wall Street Journal and The Washington Post.

```
data("usnews", package = "sentometrics")
```

ID	DATE	TEXT	WSJ	WAPO	ECONOMY	NONECONOMY
1	1995-01-02	Full text 1	1	0	1	0
2	1995-01-05	Full text 2	0	1	1	0
…	…	…	…	…	…	…

Features: relevance/importance indicators & selectors.

Massage the corpus

Checking the requirements of the corpus.

```
corpusAll <- sento_corpus(usnews)
```

Subsetting the corpus, using the *quanteda* package.

```
corpus <- quanteda::corpus_subset(corpusAll, date < "2014-10-01")
```

Adding features (for example: entities, topics, events).

```
regex <- c("\\bRepublic[s]?\\b|\\bDemocrat[s]?\\b|
           \\belection\\b|\\b[US|U.S.] [p|P]resident\\b|\\bwar\\b")
corpus <- add_features(corpus,
                       keywords = list(uncert = "uncertainty",
                                       uselect = regex,
                                       finance = c("\\bstock market")),
                       do.binary = TRUE,
                       do.regex = c(FALSE, TRUE, TRUE))
```

Pick the word lists for lexicon-based sentiment analysis

We have English, Dutch and French built-in word lists.

```
data("list_lexicons", package = "sentometrics")
data("list_valence_shifters", package = "sentometrics")
```

Prepare and check the lexicons.

```
lex <- setup_lexicons(lexiconsIn = list_lexicons[c("LM_en", "HENRY_en")],
                      valenceIn = list_valence_shifters[["en"]])
```

From sentiment to time series: aggregation specs

Aggregation of the many sentiment scores…

… within documents = document-level sentiment

… across documents = time series

… across time = *smoothed* time series

} 1 time series

… across lexicons, features and time aggregation schemes } P time series

One control function to define all of this.

```
ctrAgg <- ctr_agg(howWithin = "tf-idf",
                  howDocs = "proportional",
                  howTime = c("equal_weight", "linear", "almon"),
                  do.ignoreZeros = TRUE,
                  by = "month",
                  fill = "zero",
                  lag = 12,
                  ordersAlm = 1:3,
                  do.inverseAlm = TRUE)
```

Create many sentiment time series at once

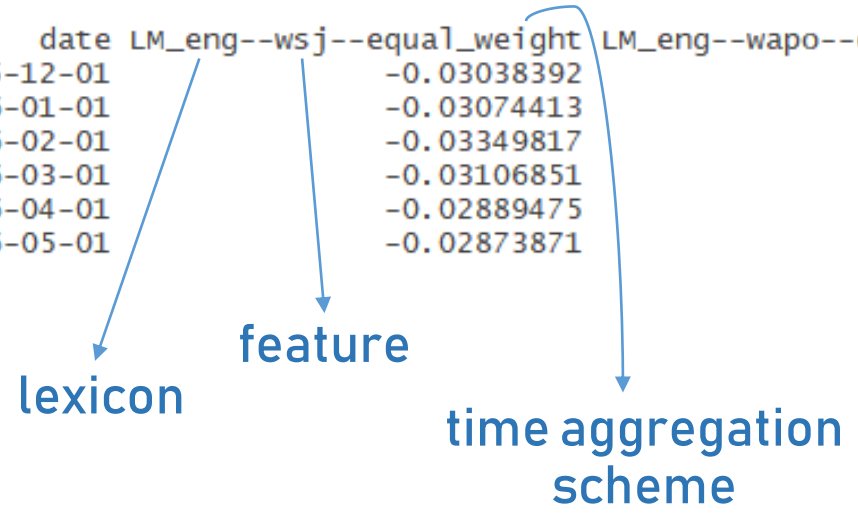
This one simple function call gives you a wide number of different sentiment time series, or "measures".

```
sentMeas <- sento_measures(corpus, lexicons = lex, ctr = ctrAgg)
```

The sentiment measures are represented as "lexicon—feature—smoothing".

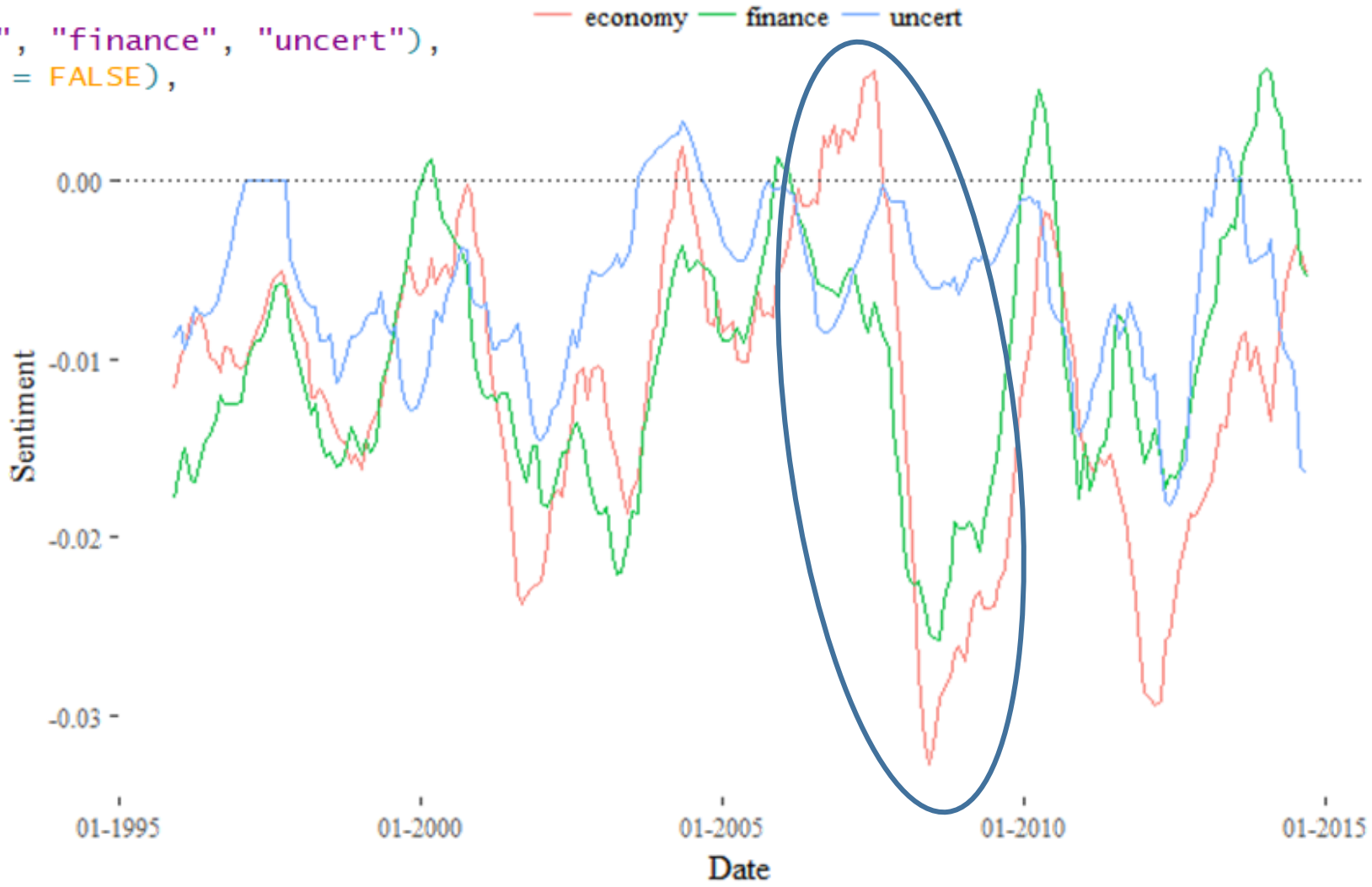
```
head(sentMeas[["measures"]][, 1:5])
```

	date	LM_eng--wsj--equal_weight	LM_eng--wapo--equal_weight	LM_eng--economy--equal_weight	LM_eng--noneconomy--equal_weight
1:	1995-12-01	-0.03038392	-0.03096058	-0.02514323	-0.03072403
2:	1996-01-01	-0.03074413	-0.03262021	-0.02200173	-0.03485245
3:	1996-02-01	-0.03349817	-0.03567584	-0.02548210	-0.03746940
4:	1996-03-01	-0.03106851	-0.03681972	-0.02363359	-0.03776122
5:	1996-04-01	-0.02889475	-0.03420715	-0.02486474	-0.03497349
6:	1996-05-01	-0.02873871	-0.03299130	-0.02532216	-0.03381545



Plot across a given time series dimension

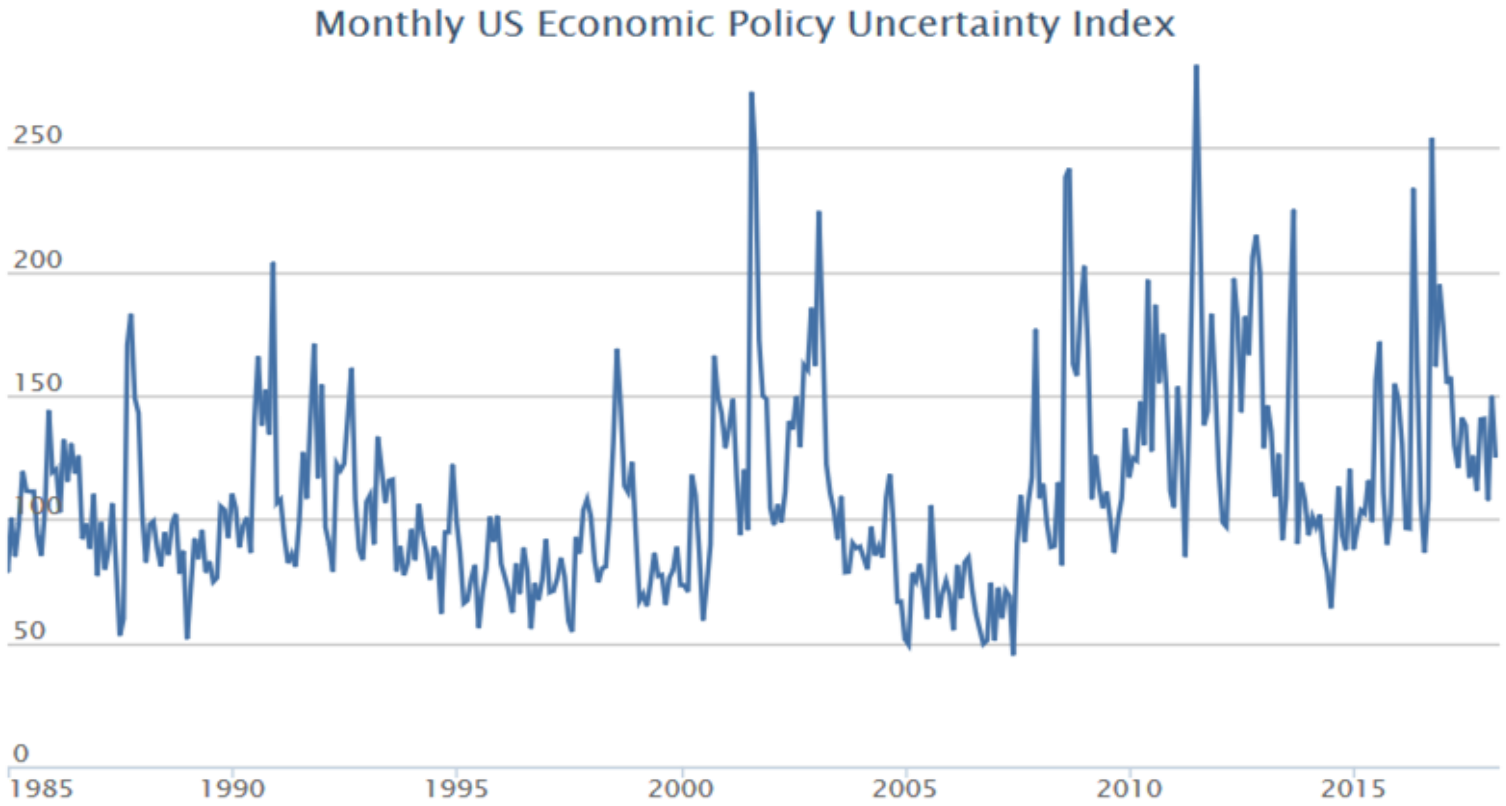
```
plot(measures_select(sentMeas,
                     c("economy", "finance", "uncert"),
                     do.combine = FALSE),
     group = "features")
```



STEPS 2–3

We try to predict the monthly U.S. EPU index…

The Economic Policy Uncertainty (EPU) index is a partly news-based measure of policy-related economic uncertainty. It is served with the package as a dataset.



http://www.policyuncertainty.com

… using elastic net regularization

We propose to use the elastic net regression (relying on *glmnet*),which balances between the LASSO and Ridge regressions through an α parameter. The large number and collinearity of the sentiment measures motivate this choice.

$$y_{u+h} = \delta + \boldsymbol{\gamma}^\top \boldsymbol{x}_u + \beta_1 s_u^1 + \ldots + \beta_p s_u^p + \ldots + \beta_P s_u^P + \epsilon_{u+h}$$

target | other explanatory variables | sentiment

A straightforward control function defines the model setup.

```
ctrIter <- ctr_model(model = "gaussian",
                     type = "BIC",
                     h = 1,
                     alphas = c(0.3, 0.5, 0.7),
                     do.iter = TRUE,
                     nSample = 36)
```

Run the prediction model iteratively

Load the data.

```
data("epu", package = "sentometrics")
y <- epu[epu[["date"]] >= sentMeas[["measures"]][["date"]][1], "index"]
```

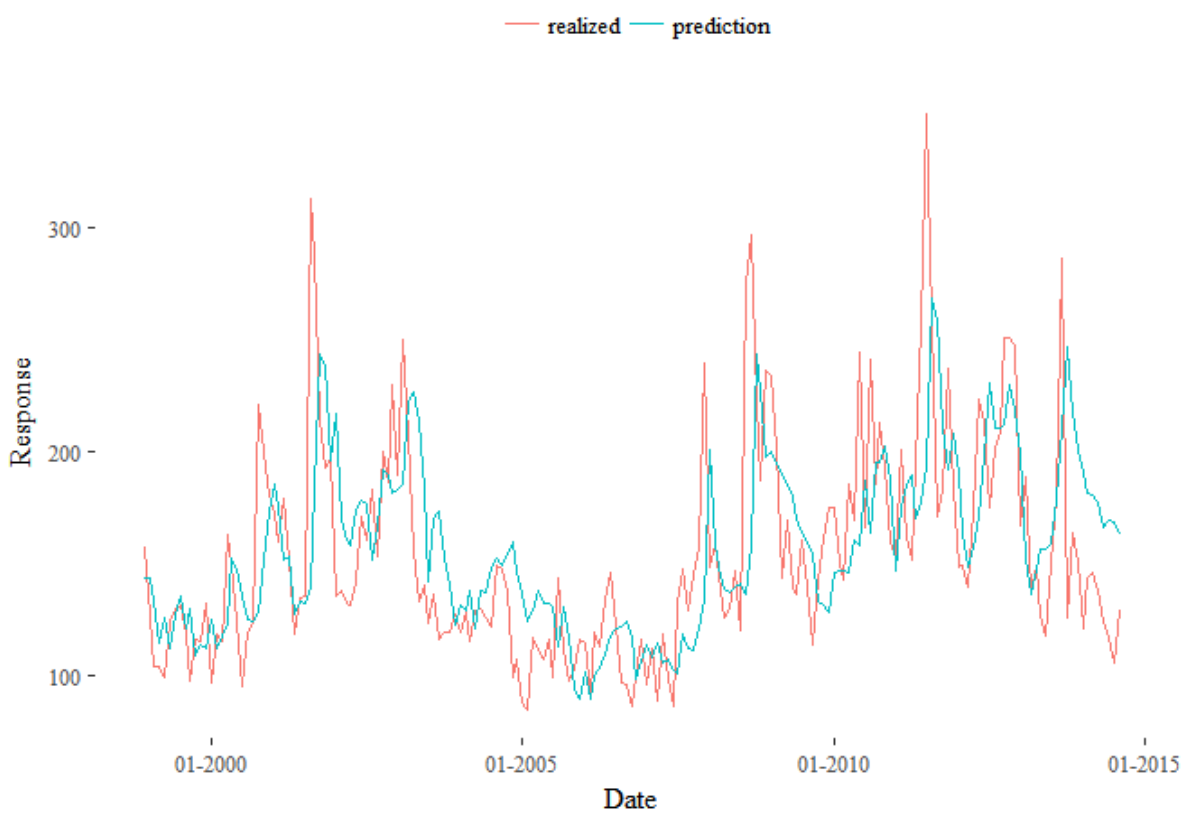
Running the out-of-sample prediction analysis is easy.

```
out <- sento_model(sentMeas, y, ctr = ctrIter)
```

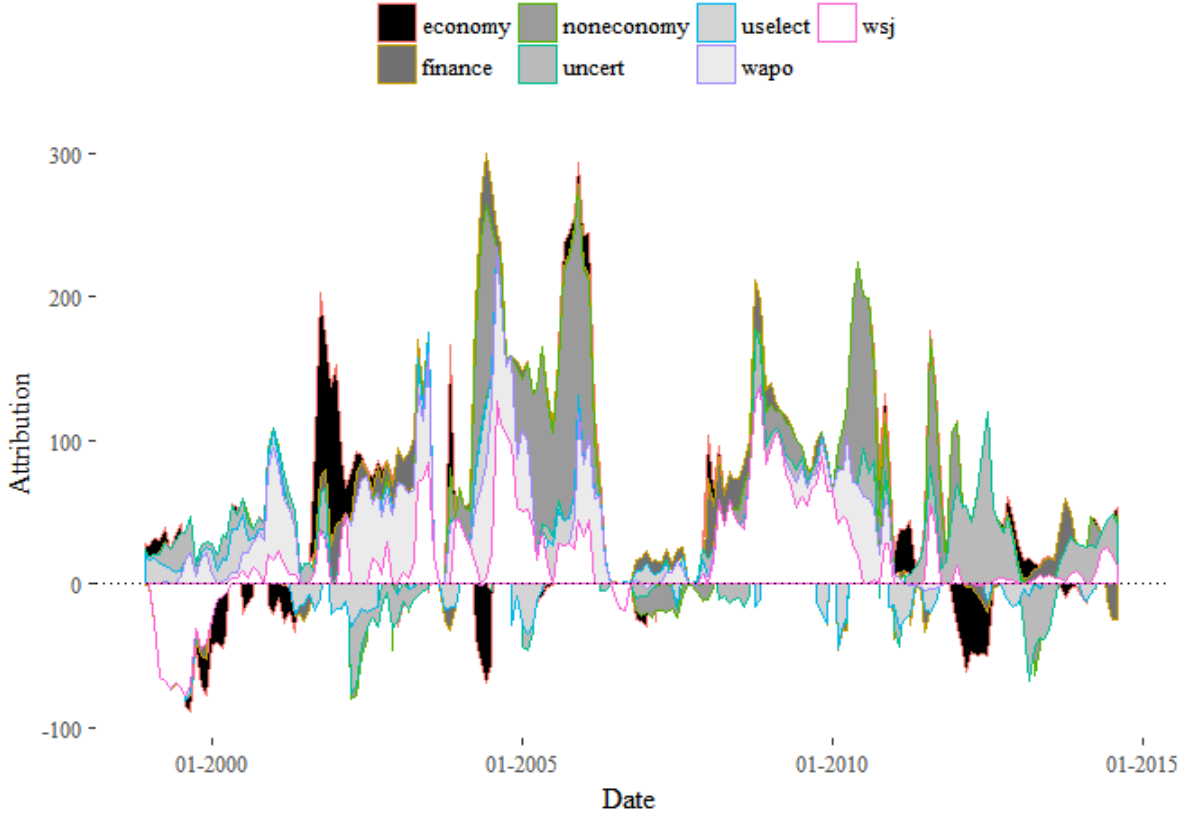
We call “attribution” the decomposition of the prediction into one of the underlying sentiment time series dimensions.

```
attr <- retrieve_attributions(out, sentMeas, do.normalize = FALSE)
```

Visualise the out-of-sample prediction and attribution




```
plot(out)
```




```
plot_attributions(attr, group = "features")
```

STEPS 4–5

Applications in finance

The package offers considerable flexibility to develop textual sentiment time series.

Many potential uses of the framework in financial and investment analysis:

- Following up of macro-trends
- Stock screening across a set of pre-defined features (e.g. sustainability)
- Abnormal sentiment detection
- Sentiment-based trading strategy
- …

Go out there and test the package!