Canada Border Services Agency
Agence des services frontaliers du Canada

Data Engineering with R, R Markdown, Shiny and algorithms

Dmitry Gorodnichy
Chief Data Office
Data Science Division

PROTECTION • SERVICE • INTEGRITY

StatsCan – Applied ML for Text Analysis CoP
25 March 2021, Ottawa

This presentation builds on the results presented at the GCData2021 conference Data Literacy Fest workshop “**Data Engineering Challenges and Solutions: Demo of Shiny”.** Please visit https://gccollab.ca/discussion/view/7407617 to view the workshop presentation/demonstration and results.

Outline

- “Round table” (Slido)

- Data Cleaning/Linking Problems: General, Canada specific
- Vision for solution: Methodological approach to Data Engineering
- Tools of the trade: algorithms, R, R Markdown, Shiny
- Use Cases: Record linking, COVID, web scraping, topic extraction
- Next steps: Introducing GCcollab **Use R!** and GCcode **r4gc** groups

- Demos & Discussion

Problems - general

• Dates :	‘20210820’ vs. ‘dob 20 Aug 2021’
• Names:	‘Dmitry Gorodnichy’ vs. ‘Dimitri Horodnytchyyi’
• Business Names:	AC, AirCanada, Air Canada Corp.
• Geographic Names:	Ottawa, Orleans, Orléans
• General Text :	“<tag> ca$h 4 u ! Sooo… C O O L! Cant believe it ☹ ”
• Postal:	“klo 0O1” vs “K100o1”
• Text matching:	Phrase matching, topics/keywords detection

Poor data quality impedes interoperability

- Good interoperability allows various data to be linked and enriched
- Probabilistic (approximate, fuzzy) matching is used to link “noisy” data
    - All words need to be compared to each other
    - Various techniques in data linkages include: using edit metrics, look-up tables, q-grams, phonetic, heuristics, …
- However, probabilistic matching has its share of challenges as well
    - How to assign threshold?
    - How to measure quality?
    - Lost nuances?
        - E.g., Bell Canada vs. Shell Canada
- No perfect solution

Problem – Canada specific

- Hundreds of dataset related to Canada
- Hundreds of data scientists working with those datasets
- Million lines of codes???
- What about quality?

Vision

- Build data-driven solution for entire GC community
  and also
  build Community of Practice
- that leverages Public Data (esp. Open Canada Data)
  and
  on Public knowledge (esp. R global community)
- so that it is
  of good quality, robust, transparent, scalable, reusable, documented, and sustainable over time

Methodological approach to Data Engineering

Software Engineering - a sub-field of Computer Science focused on developing "scientific and technological **knowledge**, **methods**, and **experience** to the design, **implementation**, **testing**, and **documentation** of <u>software</u>" [IEEE Vocabulary]

- "**Software engineering** encompasses **<u>not just the act of writing code, but all of the tools and processes</u> an organization uses to build and maintain that code over time.** …
- **Software engineering** can be thought of as "**programming integrated over time.**" [Software Engineering at Google]

Data Engineering - a sub-field of [Science and Engineering] that is focused on developing scientific and technological **knowledge, methods,** and **experience** to the design, **implementation, testing**, and **documentation** of <u>data-driven solutions</u>.

Vision (cntd)

- Stream 1: development of knowledge
  - https://gccode.ssc-spc.gc.ca/r4gc/
  - https://gccollab.ca/groups/profile/7391537/enuse-rfruse-r
  - https://github.com/canada-ca

- Stream 2: development of codes
  - R packages
  - Testbeds
  - Toolkits
  - Use cases

Taxonomy of DE tasks

Single-variable:

- General text cleaning and formatting
- Dates and time-stamps extraction, cleaning and formatting
- Canadian Postal code and municipality names recovery

Multiple-variable:

- De-duplication of entries
- Large-scale Records Linking

Relations-based:

- Entity resolution
- Text analysis and plagiarism detection

Tools

- Algorithms
- R
- R Markdown
- Shiny

Algorithms – means to automate, scale, re-use!

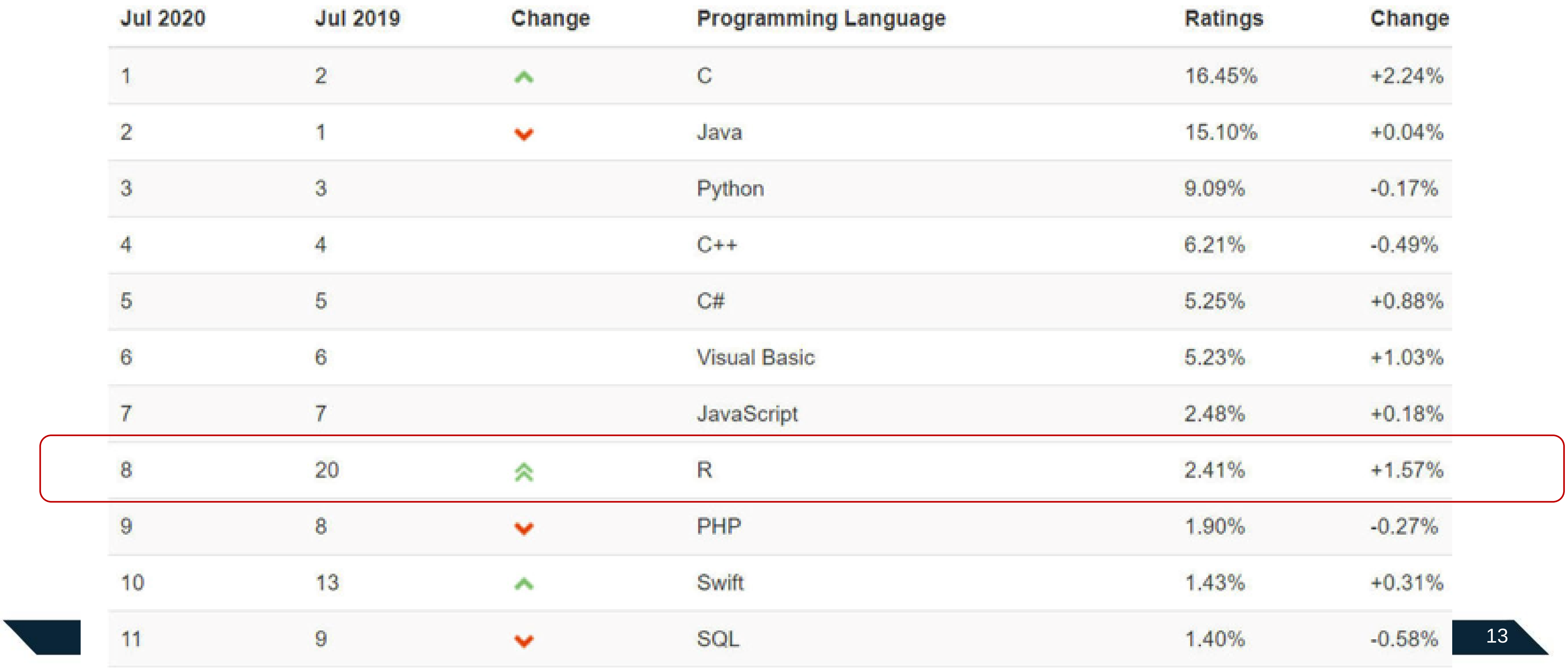
Input: **X**

(any raw “noisy” unknown data)

Output: **Y**

(meaningful “filtered” result/conclusion)

R – fastest growing in popularity environment



Jul 2020	Jul 2019	Change	Programming Language	Ratings	Change
1	2	↑	C	16.45%	+2.24%
2	1	↓	Java	15.10%	+0.04%
3	3		Python	9.09%	-0.17%
4	4		C++	6.21%	-0.49%
5	5		C#	5.25%	+0.88%
6	6		Visual Basic	5.23%	+1.03%
7	7		JavaScript	2.48%	+0.18%
8	20	⇈	R	2.41%	+1.57%
9	8	↓	PHP	1.90%	-0.27%
10	13	↑	Swift	1.43%	+0.31%
11	9	↓	SQL	1.40%	-0.58%

R Markdown – describes algorithm (from data to result)

~~~
---
title: “NLP analysis of TBS-ATI data‟
# author: ”Source: https://github.com/open-data/TBS-ATI-NLP_Exploration”
output:  html_document
---

```{r}
source(“TBS-ATI-functions.R");
# library(ATIP) # Eventually this could be converted to package or
~~~

Top 9 departments

```{r}
dtATI <- readATI()

# owners = ati%>%group_by(owner)%>% count()%>% ungroup() %>% top_n(9, n) %>% pull(owner)
aStrOwners <- dtATI[, .N, by=owner] %>% .[order(-N)] %>% .[1:9, owner]
```

...
```
```

R Shiny – enables interactive testing and dashboards

~~~
---
title: "Data Engineering Testbed‟
# author: ”Source: https://gccode.ssc-spc.gc.ca/gorodnichy/rCanada”
output:  flex_dashboard
runtime: shiny
---

```{r}
source(“rCanada-functions.R");
#library(rCanada)
dtNames <- as.data.table(lexicon::common_names ) %>% setnames("Name")
~~~

```{r de_1_dates.Rmd, child = 'de_1_dates.Rmd'}
```

```{r}
r.dtNames <- reactive({
    dtNames [, dist:=stringdist( Name, input$typedName, input$metric] })
```

...
```
```

Use Cases

- Records deduplication and linking: https://rCanada.shinyapps.io/demo
- Web crawling: …/demo/#section-web-crawling
    - Dates extraction
    - Finding nicknames and names variants

- UofT COVID data: https://rCanada.shinyapps.io/covid
- CBSA BWT data: https://itrack.shinyapps.io/border
- TBS PSES data: https://itrack.shinyapps.io/PSES
- TBS ATIP data: https://rCanada.shinyapps.io/TBS-ATI-NLP

Steps for Record Linking / Deduplication

1. Data preparation: feature selection and preparation
2. Perform pair-wise comparison
3. Set constraints:
   soft vs. hard constraints, inter- vs. intra- class relationships
4. String similarity metrics (stringsim):
   q-grams vs. edit steps vs. heuristics vs. soundex
5. Algorithms:
   automated vs. semi-automated
6. Quality & Precision metrics: Accuracy vs. Precision/Recall

Ref:

Steps for Text Analysis / Topic Extraction

- Load thesaurus and stop-words
- Words as token / unigrams
- Compute Top words, N/Total, bigram and n-grams
- Compute TF-IDF (term frequency – inverse document frequency)
- Compute correlations
    - Visualize bigram / n-grams relationship (ggraph, wordcloud)
- Topic modeling (w. topicmodel / textmineR):
    - Compute DTM (document term matrix)
    - Compute LDA (Latent Dirichlet Allocation)
    - Visualize dominant topics (ggplot, wordcloud)

Refs: S. Silge, D, Robinson,“Text Mining with R: a Tidy Approach”, tidytextmining.com (github.com/dgrtwo/tidy-text-mining)
https://gccollab.ca/discussion/view/7404441/text-analysis-in-r

Next steps

- The works has just started. Much more ahead.
  - GCcollab group “Use R!”
  - GCcode group “r4gc”: https://gccode.ssc-spc.gc.ca/r4gc/
  - Codes, Apps
- We need your help!
  - curating DE challenges and public domain solutions (codes/papers)
  - curating public domain Data-sets
  - testing & benchmarking
- Join GCcollab / GCcode groups.
- Join Friday “Lunch and Learn R” meet-ups
- Contact: Dmitry.Gorodnichy@cbsa-asfc.gc.ca

Time for Demo and Discussion!

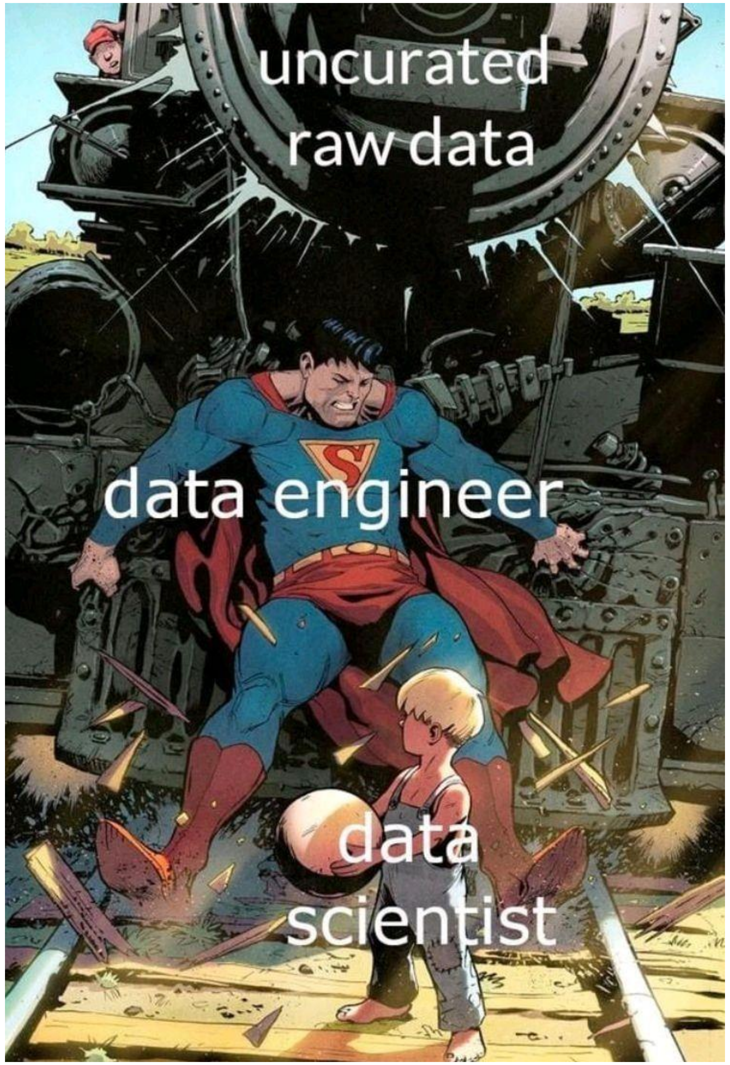
Acknowledgements: The author gratefully acknowledges the use of the following public domain material:

R packages / Codes: data.table, dtplyr, lubridate, magrittr; tidytext, antiword, filehash, textreg, textreuse, stringr, stringi, textclean, syuzhet, hunspell, textshape; soundex, phonetic; RecordLinkage, stringdist, blink, reclin, fuzzyjoin, fastlink; shiny, rsconnect, knitr, rmarkdown, flexdashboard, DT; cancensus; googleway, tidygeocoder, geocompr; rvest, httr, xml2, jsonlite; microbenchmark.; NLP, quanteda, udpipe, spacyr, tidytext; qdap, tm, lexicon, Rnewsflow, textcat; ggraph, widyr,tm,topicmodels,textmineR,ggwordcloud, wordcloud, github.com/open-data

Data: Statistics Canada, Post Canada, Simple Maps, TBS ATI data and codes by Patrick Little (TBS)

Image: by Anna Nyulund (LinkedIn post)

Discussions with many CBSA and GC colleagues, in particular at Friday's meet-ups.

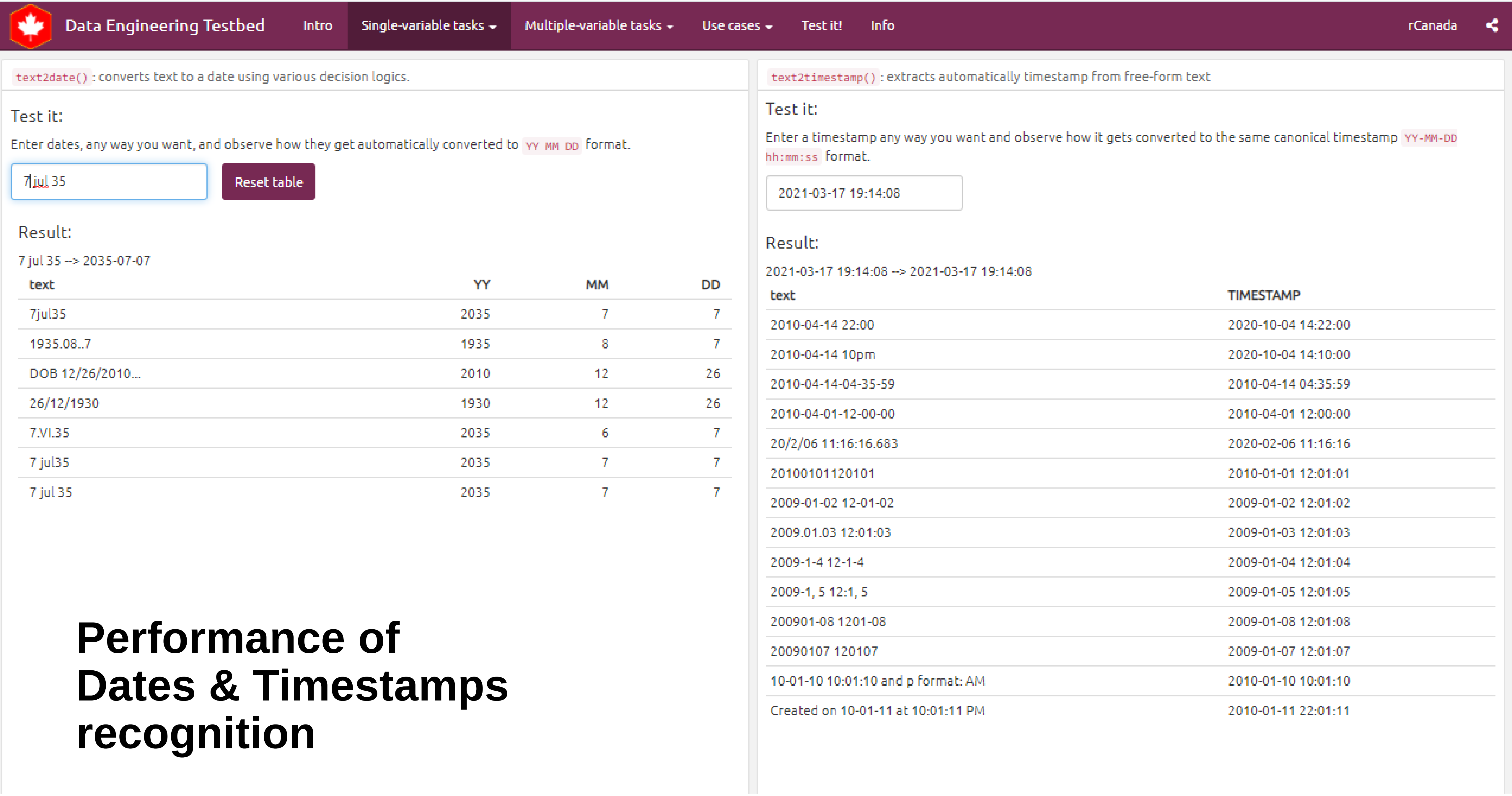


Appendices

- Records cleaning, deduplication and linking:
  https://rCanada.shinyapps.io/demo
  (Leveraging various R packages for data cleaning and linking)

- NLP topic modeling in TBS ATIP data:
  https://rCanada.shinyapps.io/TBS-ATI-NLP
  (Leveraging the work of TBS and various R packages for text mining)

Performance of Dates & Timestamps recognition



`text2date()` : converts text to a date using various decision logics.

Test it:

Enter dates, any way you want, and observe how they get automatically converted to `YY MM DD` format.

7 jul 35

Result:

7 jul 35 --> 2035-07-07

text	YY	MM	DD
7jul35	2035	7	7
1935.08..7	1935	8	7
DOB 12/26/2010...	2010	12	26
26/12/1930	1930	12	26
7.VI.35	2035	6	7
7 jul35	2035	7	7
7 jul 35	2035	7	7

`text2timestamp()` : extracts automatically timestamp from free-form text

Test it:

Enter a timestamp any way you want and observe how it gets converted to the same canonical timestamp `YY-MM-DD hh:mm:ss` format.

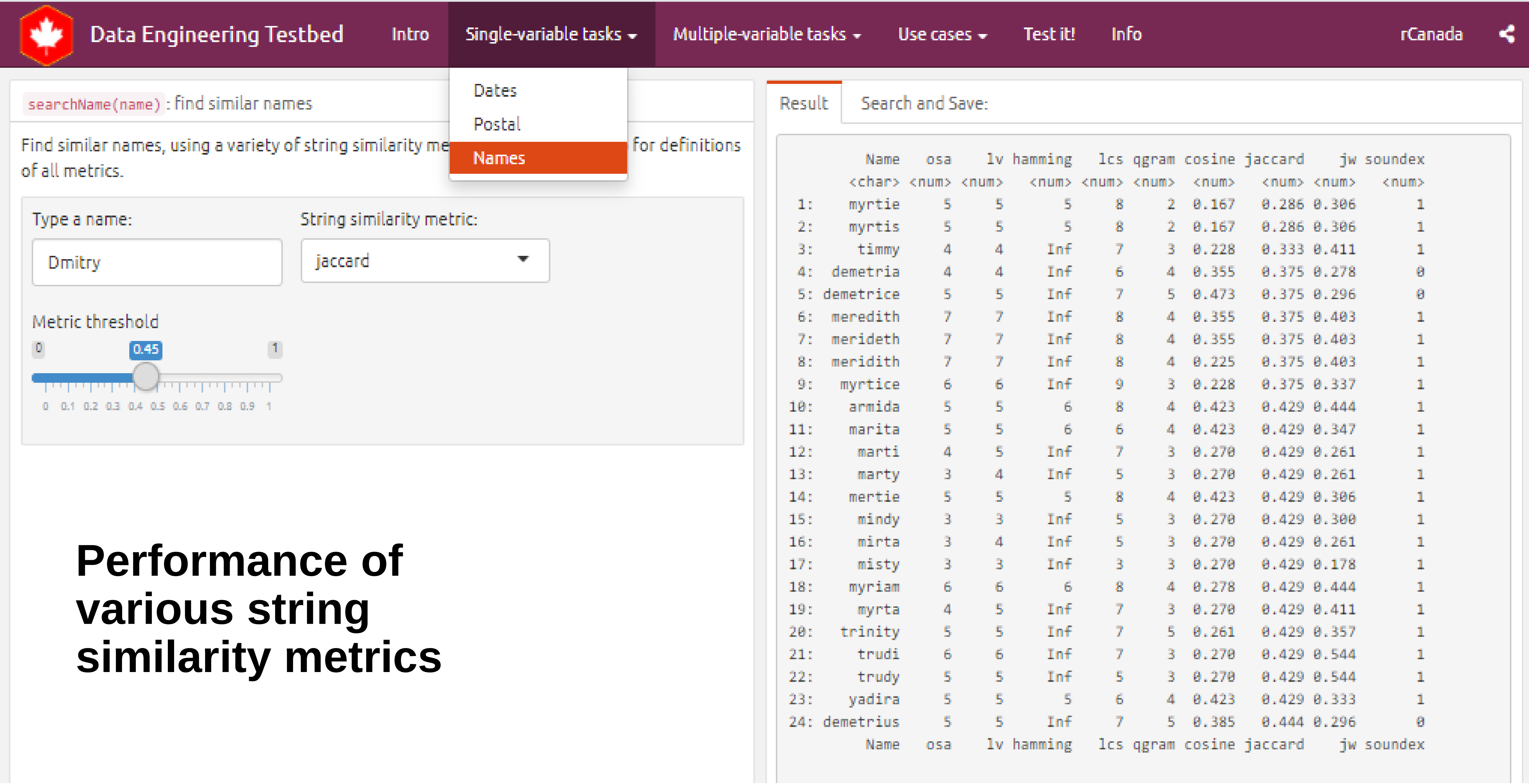
2021-03-17 19:14:08

Result:

2021-03-17 19:14:08 --> 2021-03-17 19:14:08

text	TIMESTAMP
2010-04-14 22:00	2020-10-04 14:22:00
2010-04-14 10pm	2020-10-04 14:10:00
2010-04-14-04-35-59	2010-04-14 04:35:59
2010-04-01-12-00-00	2010-04-01 12:00:00
20/2/06 11:16:16.683	2020-02-06 11:16:16
20100101120101	2010-01-01 12:01:01
2009-01-02 12-01-02	2009-01-02 12:01:02
2009.01.03 12:01:03	2009-01-03 12:01:03
2009-1-4 12-1-4	2009-01-04 12:01:04
2009-1, 5 12:1, 5	2009-01-05 12:01:05
200901-08 1201-08	2009-01-08 12:01:08
20090107 120107	2009-01-07 12:01:07
10-01-10 10:01:10 and p format: AM	2010-01-10 10:01:10
Created on 10-01-11 at 10:01:11 PM	2010-01-11 22:01:11

Performance of various string similarity metrics



`searchName(name)` : find similar names

Find similar names, using a variety of string similarity me for definitions of all metrics.

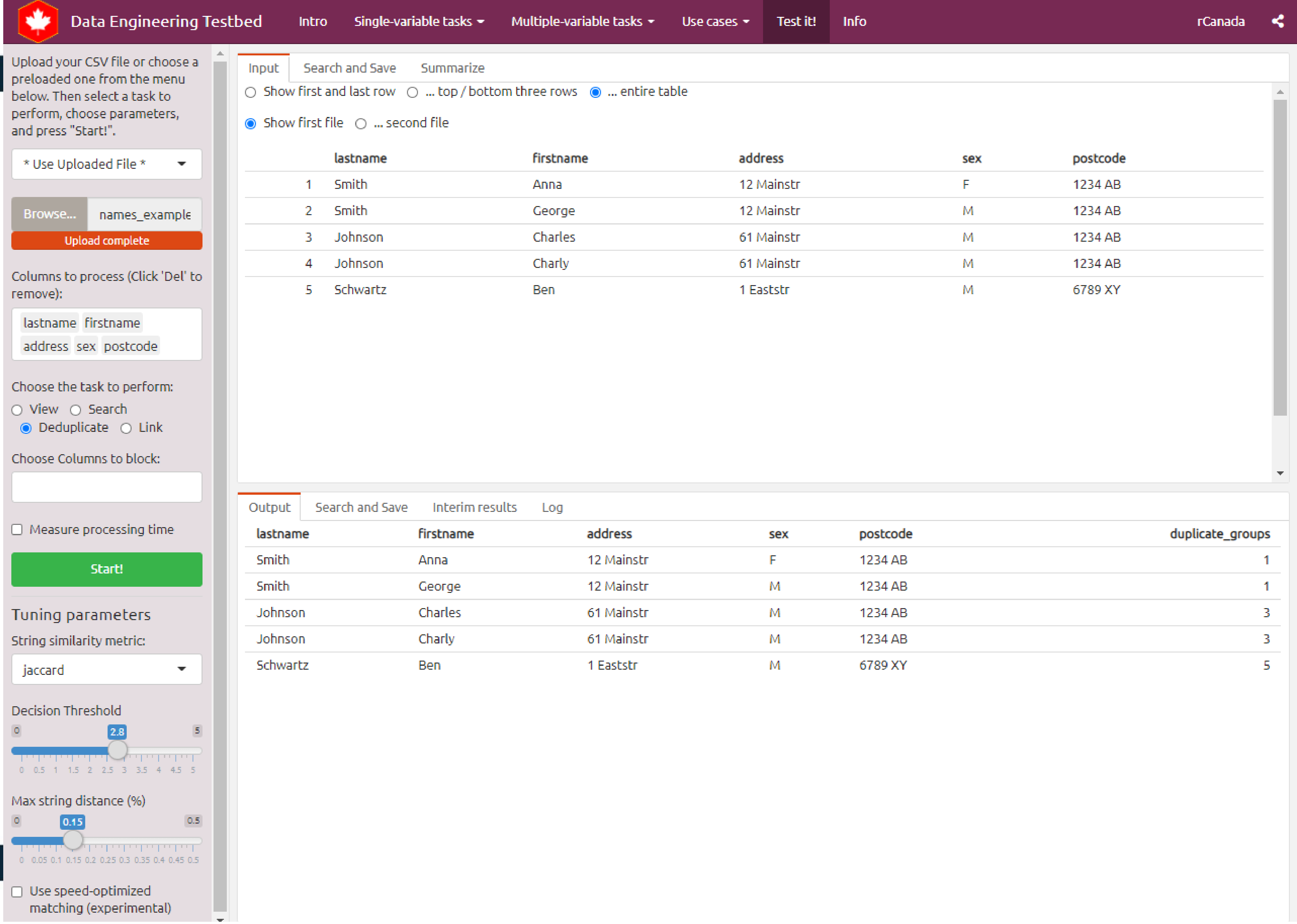
Type a name: Dmitry

String similarity metric: jaccard

Metric threshold: 0.45

```
          Name   osa    lv hamming   lcs qgram cosine jaccard    jw soundex
        <char> <num> <num>   <num> <num> <num>  <num>   <num> <num>   <num>
 1:     myrtie     5     5       5     8     2  0.167   0.286 0.306       1
 2:     myrtis     5     5       5     8     2  0.167   0.286 0.306       1
 3:      timmy     4     4     Inf     7     3  0.228   0.333 0.411       1
 4:   demetria     4     4     Inf     6     4  0.355   0.375 0.278       0
 5:  demetrice     5     5     Inf     7     5  0.473   0.375 0.296       0
 6:   meredith     7     7     Inf     8     4  0.355   0.375 0.403       1
 7:   merideth     7     7     Inf     8     4  0.355   0.375 0.403       1
 8:   meridith     7     7     Inf     8     4  0.225   0.375 0.403       1
 9:    myrtice     6     6     Inf     9     3  0.228   0.375 0.337       1
10:     armida     5     5       6     8     4  0.423   0.429 0.444       1
11:     marita     5     5       6     6     4  0.423   0.429 0.347       1
12:      marti     4     5     Inf     7     3  0.270   0.429 0.261       1
13:      marty     3     4     Inf     5     3  0.270   0.429 0.261       1
14:     mertie     5     5       5     8     4  0.423   0.429 0.306       1
15:      mindy     3     3     Inf     5     3  0.270   0.429 0.300       1
16:      mirta     3     4     Inf     5     3  0.270   0.429 0.261       1
17:      misty     3     3     Inf     3     3  0.270   0.429 0.178       1
18:     myriam     6     6       6     8     4  0.278   0.429 0.444       1
19:      myrta     4     5     Inf     7     3  0.270   0.429 0.411       1
20:    trinity     5     5     Inf     7     5  0.261   0.429 0.357       1
21:      trudi     6     6     Inf     7     3  0.270   0.429 0.544       1
22:      trudy     5     5     Inf     5     3  0.270   0.429 0.544       1
23:     yadira     5     5       5     6     4  0.423   0.429 0.333       1
24:  demetrius     5     5     Inf     7     5  0.385   0.444 0.296       0
          Name   osa    lv hamming   lcs qgram cosine jaccard    jw soundex
```

Record deduplication



Upload your CSV file or choose a preloaded one from the menu below. Then select a task to perform, choose parameters, and press "Start!".

* Use Uploaded File *

names_example

Upload complete

Columns to process (Click 'Del' to remove): lastname firstname address sex postcode

Choose the task to perform:

○ View ○ Search ● Deduplicate ○ Link

Choose Columns to block:

□ Measure processing time

Tuning parameters

String similarity metric: jaccard

Decision Threshold: 2.8 (0 – 5)

Max string distance (%): 0.15 (0 – 0.5)

□ Use speed-optimized matching (experimental)

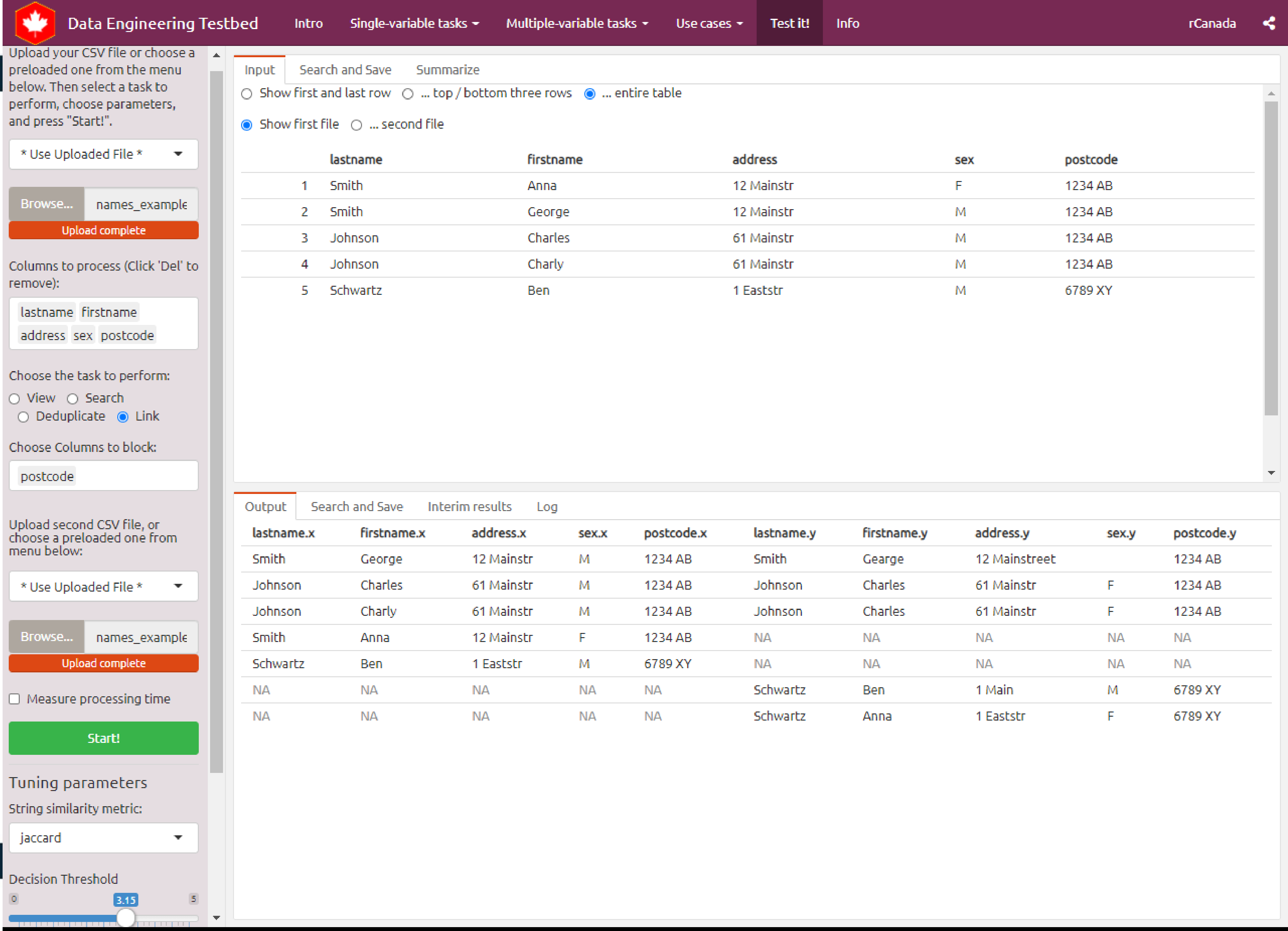
○ Show first and last row ○ ... top / bottom three rows ● ... entire table

● Show first file ○ ... second file

	lastname	firstname	address	sex	postcode
1	Smith	Anna	12 Mainstr	F	1234 AB
2	Smith	George	12 Mainstr	M	1234 AB
3	Johnson	Charles	61 Mainstr	M	1234 AB
4	Johnson	Charly	61 Mainstr	M	1234 AB
5	Schwartz	Ben	1 Eaststr	M	6789 XY

lastname	firstname	address	sex	postcode	duplicate_groups
Smith	Anna	12 Mainstr	F	1234 AB	1
Smith	George	12 Mainstr	M	1234 AB	1
Johnson	Charles	61 Mainstr	M	1234 AB	3
Johnson	Charly	61 Mainstr	M	1234 AB	3
Schwartz	Ben	1 Eaststr	M	6789 XY	5

Record linking



Upload your CSV file or choose a preloaded one from the menu below. Then select a task to perform, choose parameters, and press "Start!".

* Use Uploaded File *

Browse... names_example

Upload complete

Columns to process (Click 'Del' to remove):

lastname firstname address sex postcode

Choose the task to perform:

○ View ○ Search
○ Deduplicate ● Link

Choose Columns to block:

postcode

Upload second CSV file, or choose a preloaded one from menu below:

* Use Uploaded File *

Browse... names_example

Upload complete

□ Measure processing time

Start!

Tuning parameters

String similarity metric:

jaccard

Decision Threshold

0 3.15 5

○ Show first and last row ○ ... top / bottom three rows ● ... entire table

● Show first file ○ ... second file

	lastname	firstname	address	sex	postcode
1	Smith	Anna	12 Mainstr	F	1234 AB
2	Smith	George	12 Mainstr	M	1234 AB
3	Johnson	Charles	61 Mainstr	M	1234 AB
4	Johnson	Charly	61 Mainstr	M	1234 AB
5	Schwartz	Ben	1 Eaststr	M	6789 XY

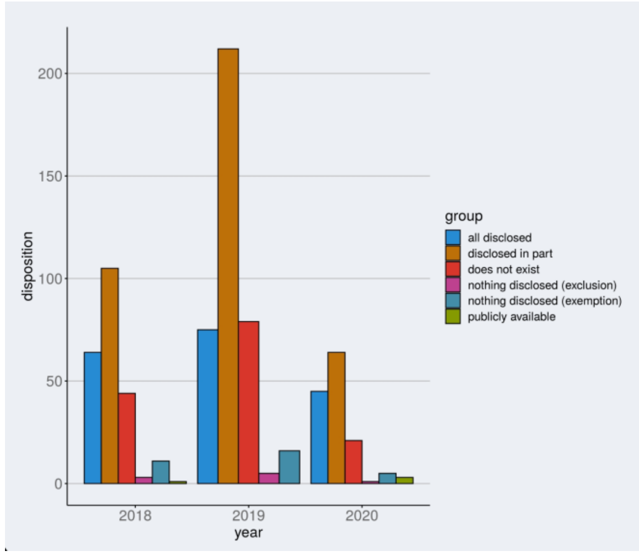
lastname.x	firstname.x	address.x	sex.x	postcode.x	lastname.y	firstname.y	address.y	sex.y	postcode.y
Smith	George	12 Mainstr	M	1234 AB	Smith	Gearge	12 Mainstreet		1234 AB
Johnson	Charles	61 Mainstr	M	1234 AB	Johnson	Charles	61 Mainstr	F	1234 AB
Johnson	Charly	61 Mainstr	M	1234 AB	Johnson	Charles	61 Mainstr	F	1234 AB
Smith	Anna	12 Mainstr	F	1234 AB	NA	NA	NA	NA	NA
Schwartz	Ben	1 Eaststr	M	6789 XY	NA	NA	NA	NA	NA
NA	NA	NA	NA	NA	Schwartz	Ben	1 Main	M	6789 XY
NA	NA	NA	NA	NA	Schwartz	Anna	1 Eaststr	F	6789 XY

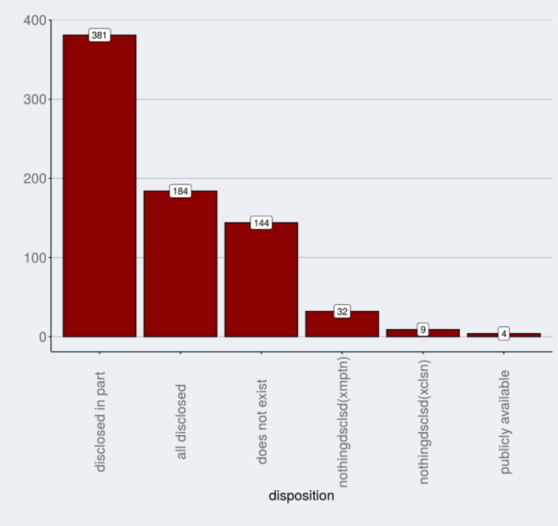
Appendices

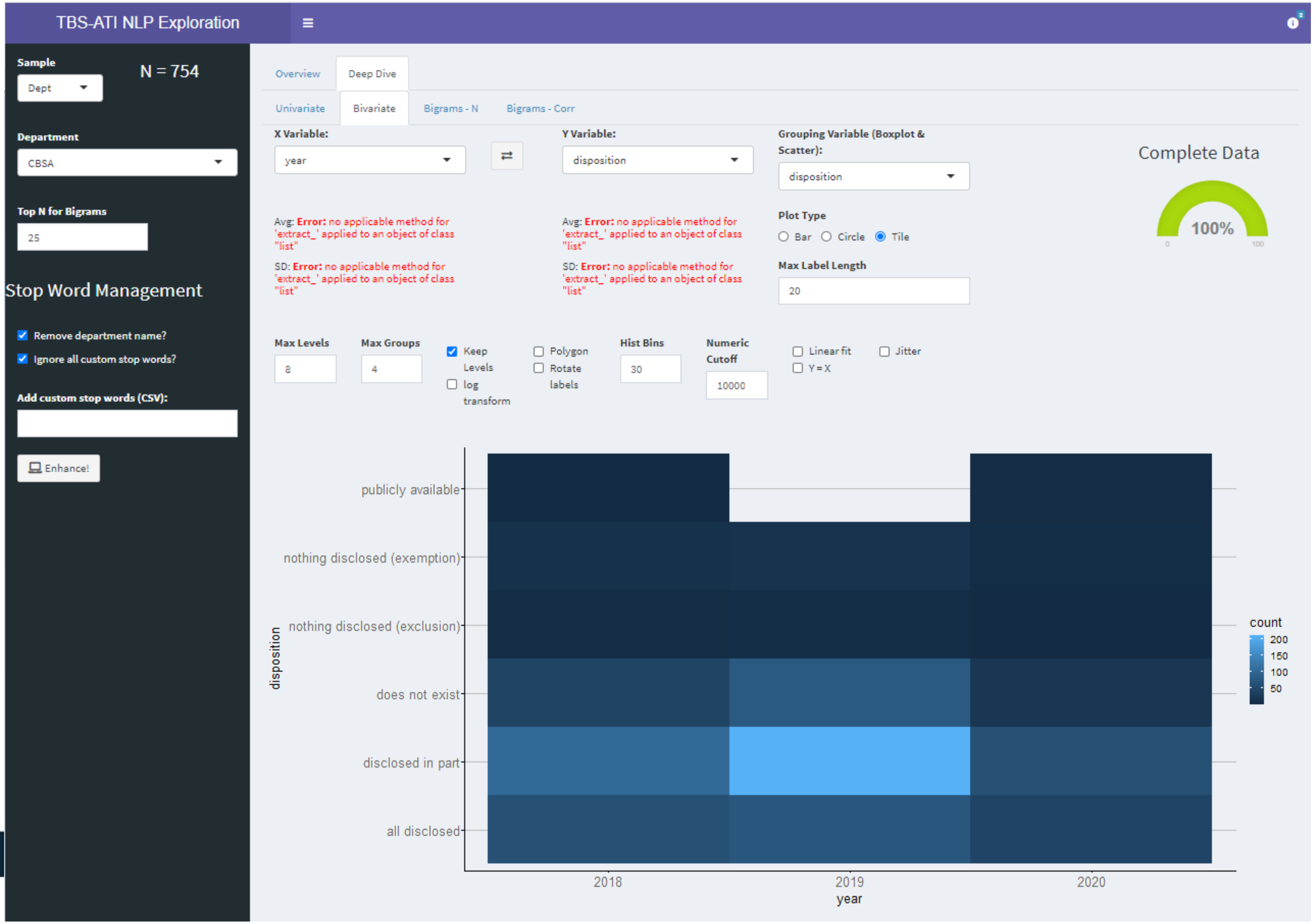
- Records cleaning, deduplication and linking:
  https://rCanada.shinyapps.io/demo
  (Leveraging various R packages for data cleaning and linking)

- NLP topic modeling in TBS ATIP data:
  https://rCanada.shinyapps.io/TBS-ATI-NLP
  (Leveraging the work of TBS and various R packages for text mining)

Univariate and bivariate analysis of dataset variables







Sample: Dept

N = 754

Department: CBSA

Top N for Bigrams: 25

Stop Word Management

☑ Remove department name?

☑ Ignore all custom stop words?

Add custom stop words (CSV):

X Variable: year

Y Variable: disposition

Grouping Variable (Boxplot & Scatter): disposition

Avg: **Error:** no applicable method for 'extract_' applied to an object of class "list"

SD: **Error:** no applicable method for 'extract_' applied to an object of class "list"

Avg: **Error:** no applicable method for 'extract_' applied to an object of class "list"

SD: **Error:** no applicable method for 'extract_' applied to an object of class "list"

Plot Type: ○ Bar ○ Circle ● Tile

Max Label Length: 20

Max Levels: 8

Max Groups: 4

☑ Keep Levels

□ log transform

□ Polygon

□ Rotate labels

Hist Bins: 30

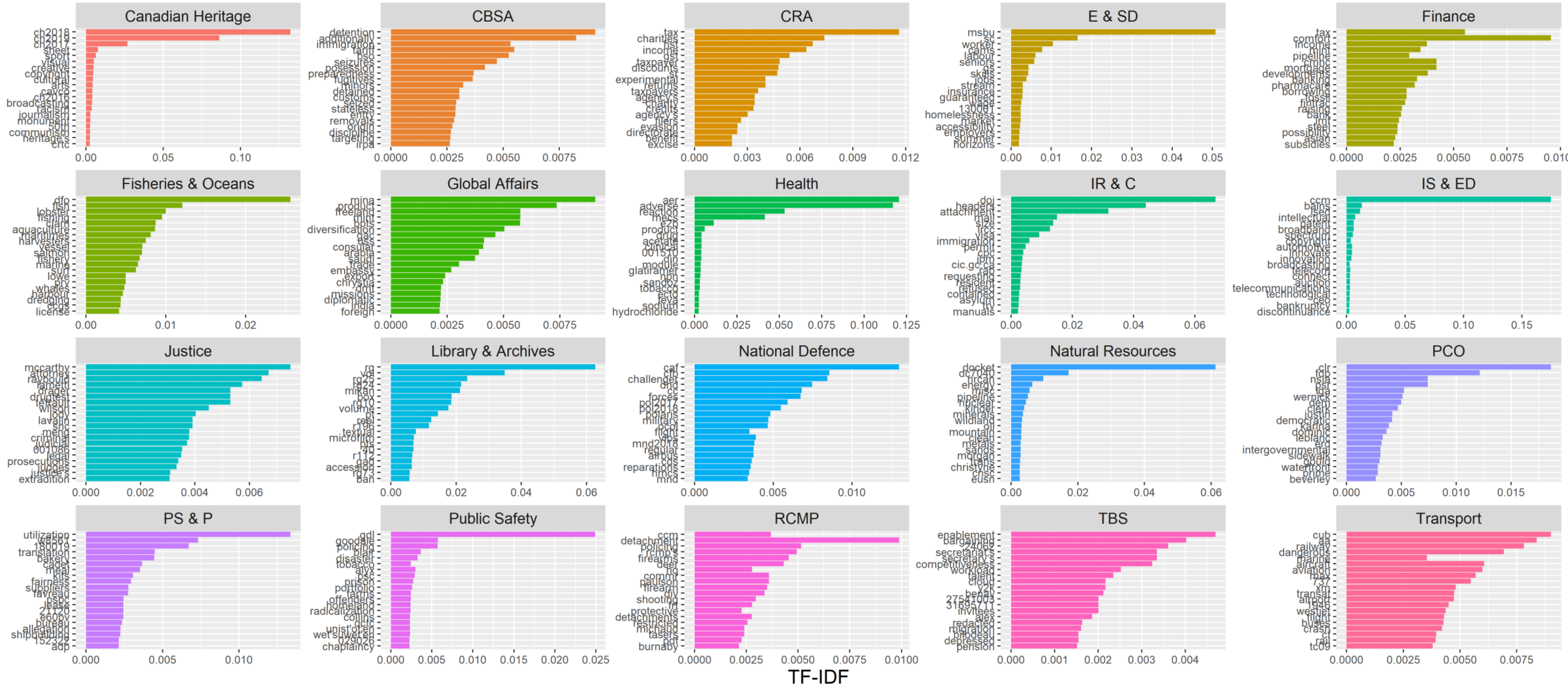
Numeric Cutoff: 10000

□ Linear fit

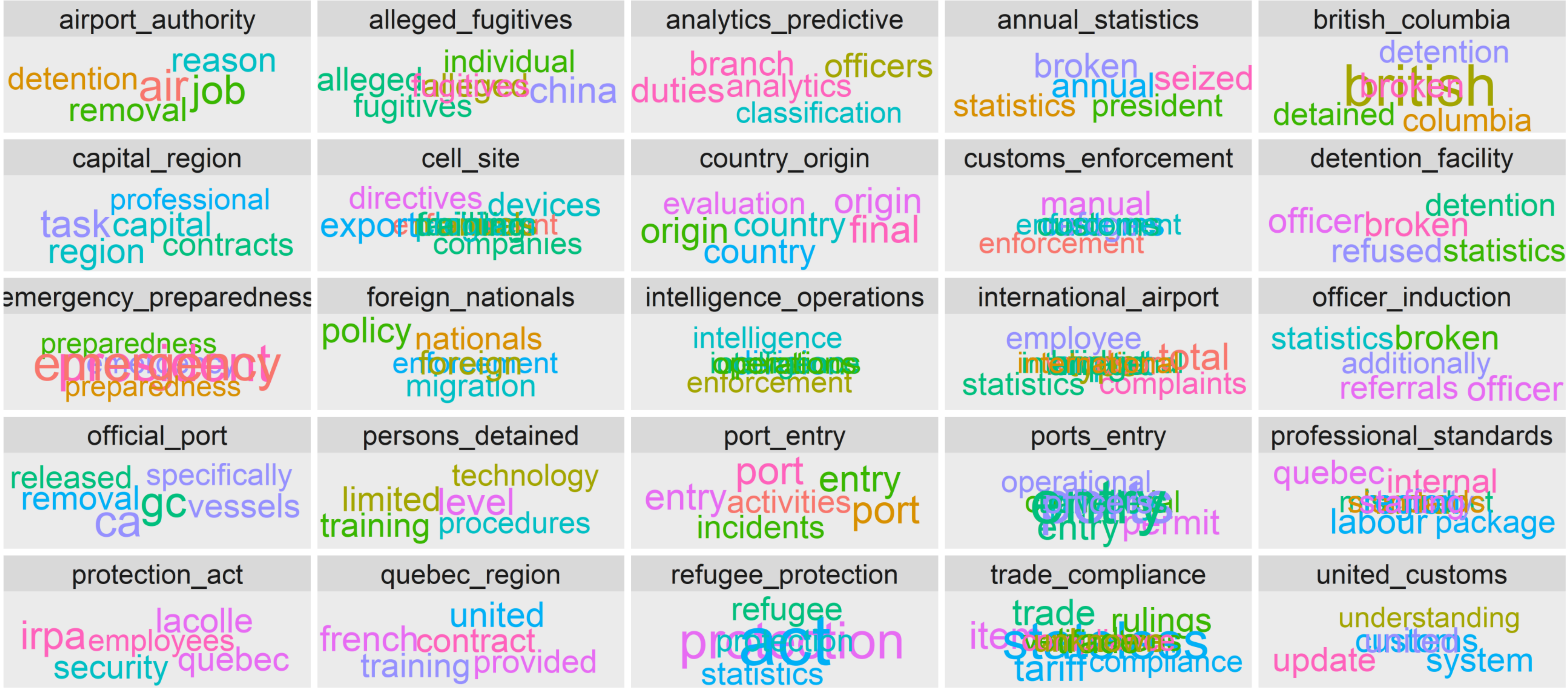
□ Y=X

□ Jitter

1-grams (single words)

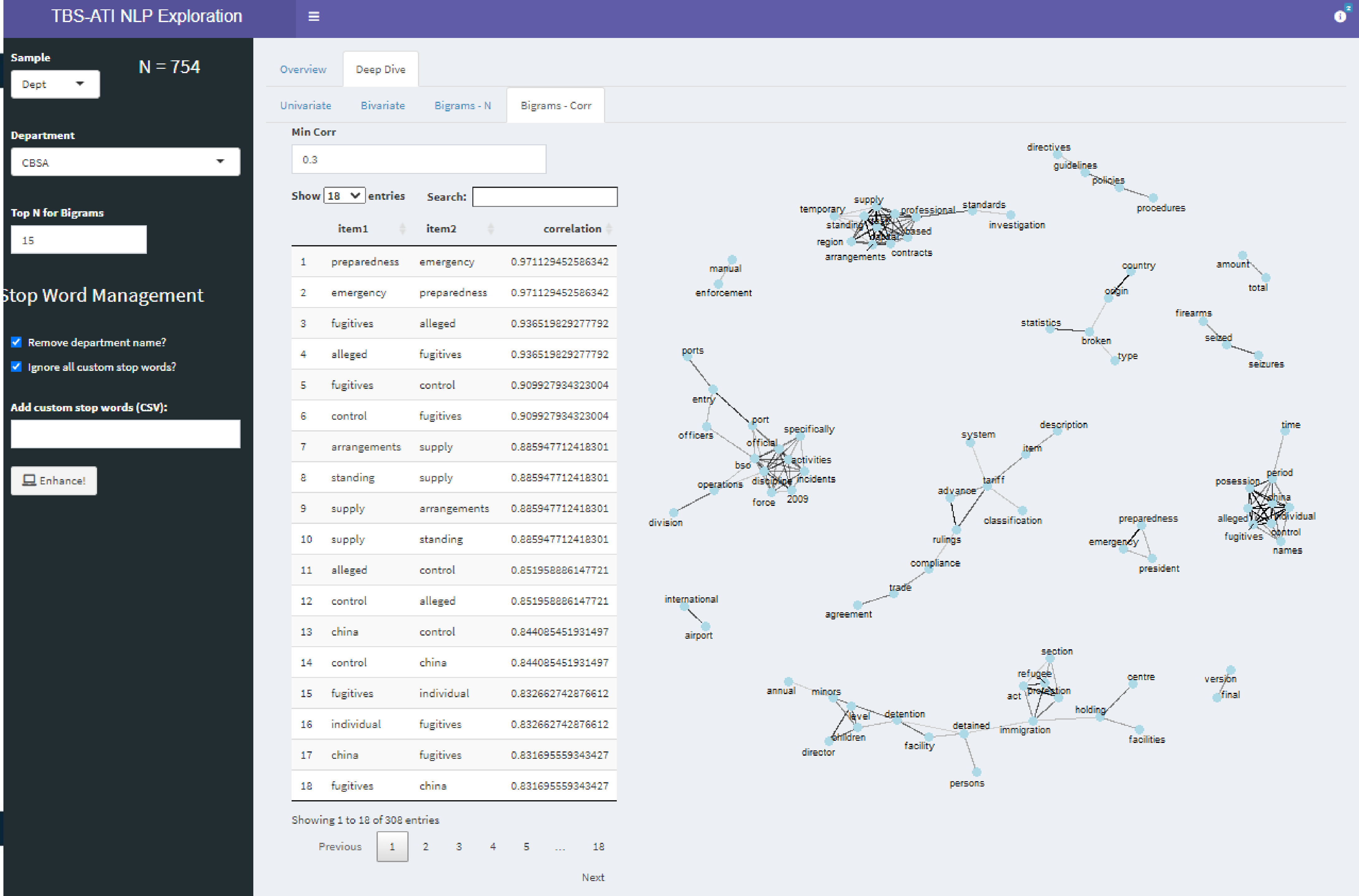


Topic modeling (30 main topics): wordcloud



Topic modelling:

Graph / Network view



Sample: Dept

N = 754

Department: CBSA

Top N for Bigrams: 15

Stop Word Management

☑ Remove department name?

☑ Ignore all custom stop words?

Add custom stop words (CSV):

Min Corr: 0.3

Show 18 entries Search:

	item1	item2	correlation
1	preparedness	emergency	0.971129452586342
2	emergency	preparedness	0.971129452586342
3	fugitives	alleged	0.936519829277792
4	alleged	fugitives	0.936519829277792
5	fugitives	control	0.909927934323004
6	control	fugitives	0.909927934323004
7	arrangements	supply	0.885947712418301
8	standing	supply	0.885947712418301
9	supply	arrangements	0.885947712418301
10	supply	standing	0.885947712418301
11	alleged	control	0.851958886147721
12	control	alleged	0.851958886147721
13	china	control	0.844085451931497
14	control	china	0.844085451931497
15	fugitives	individual	0.832662742876612
16	individual	fugitives	0.832662742876612
17	china	fugitives	0.831695559343427
18	fugitives	china	0.831695559343427

Showing 1 to 18 of 308 entries