Ministry of Justice

Coffee & Coding

Splink - *data linkage at scale*

Theodore Manassis, Sam Lindsay
22nd July 2020

Protecting and advancing the principles of justice

Outline

- Business problem
- Technical challenges
- Existing solutions + why we decided to build **splink**
- Data linkage basics
- Fellegi-Sunter theory
- Splink demo
- Further advice for getting started

Business problem

- Many different large-scale sources of administrative data

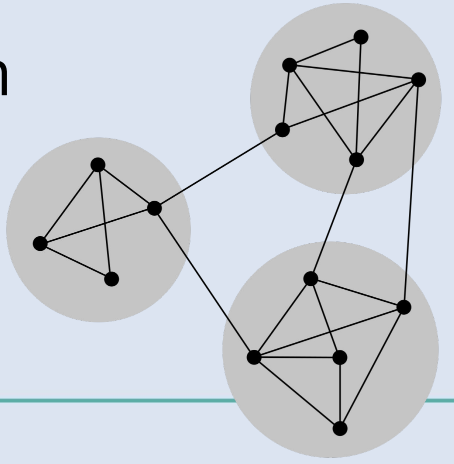
- No single unique identifier

- Inconsistent use of existing identifiers

- Data linking will underpin improved insights effectiveness of justice system interventions and repeat service users

Technical challenges

- Tens of millions of records
- Data sources typically need to be both linked and deduplicated
- Lack of large-scale, realistic training datasets
- Given variety of input datasets, need a very flexible solution
- Transitive linking and resolving the graph



Existing tools

- We reviewed available open source software, concentrating on packages in **R**, **Python** and **Spark**, because these are the main tools available on our Analytical Platform

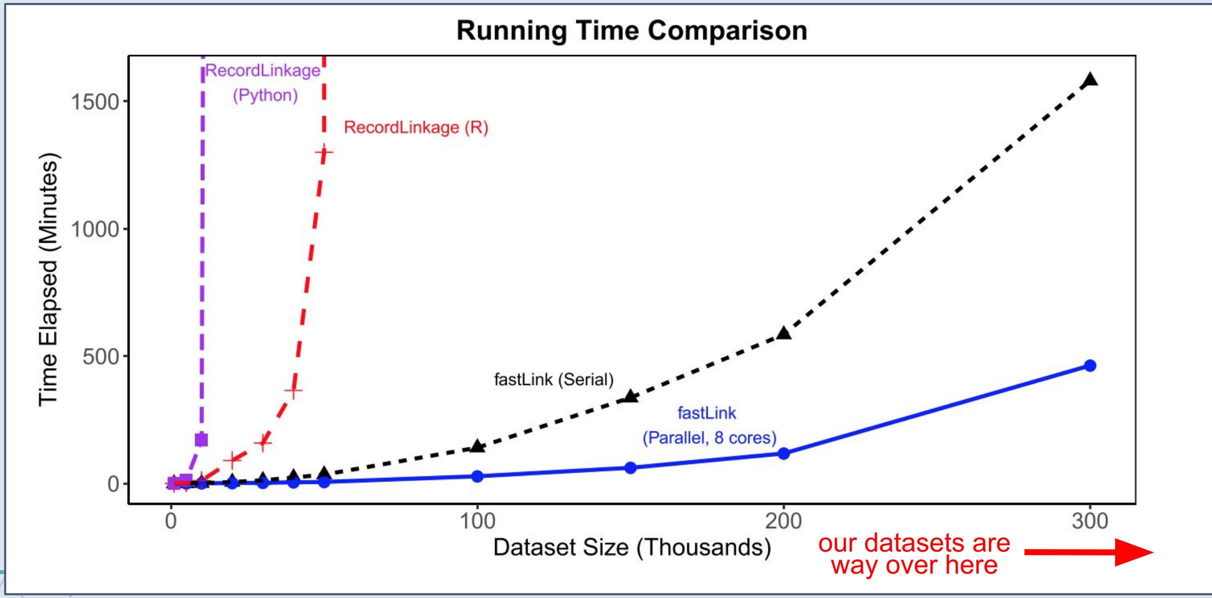
- R FastLink performs comparatively well, and is rigorous, with a formal academic paper supporting its methodology

American Political Science Review (2019) 113, 2, 353–371
doi:10.1017/S0003055418000783 © American Political Science Association 2019

Using a Probabilistic Model to Assist Merging of Large-Scale Administrative Records

TED ENAMORADO *Princeton University*
BENJAMIN FIFIELD *Princeton University*
KOSUKE IMAI *Harvard University*

- However, record linkage not really suited to in-memory computations



Why did we decide to build a new package?

- No existing open source software which works at the necessary scale.
- R’s FastLink package probably best current implementation.
- Fellegi-Sunter/Expectation Maximisation (FS/EM) methodology offers good balance of transparency and performance
- FS/EM methodology almost “trivially parallelisable” so very suited to distributed computing frameworks like Apache Spark

Data Linkage basics (in brief)

Data quality sometimes can be problematic

Type of error	Data Entry method
Typographical	Keyboard
Phonetic	Dictation
OCR error	Optical Character Recognition of handwritten material

Data Linkage basics (in brief)

Matching records

- Manual:

  Clerical (not feasible for large datasets)

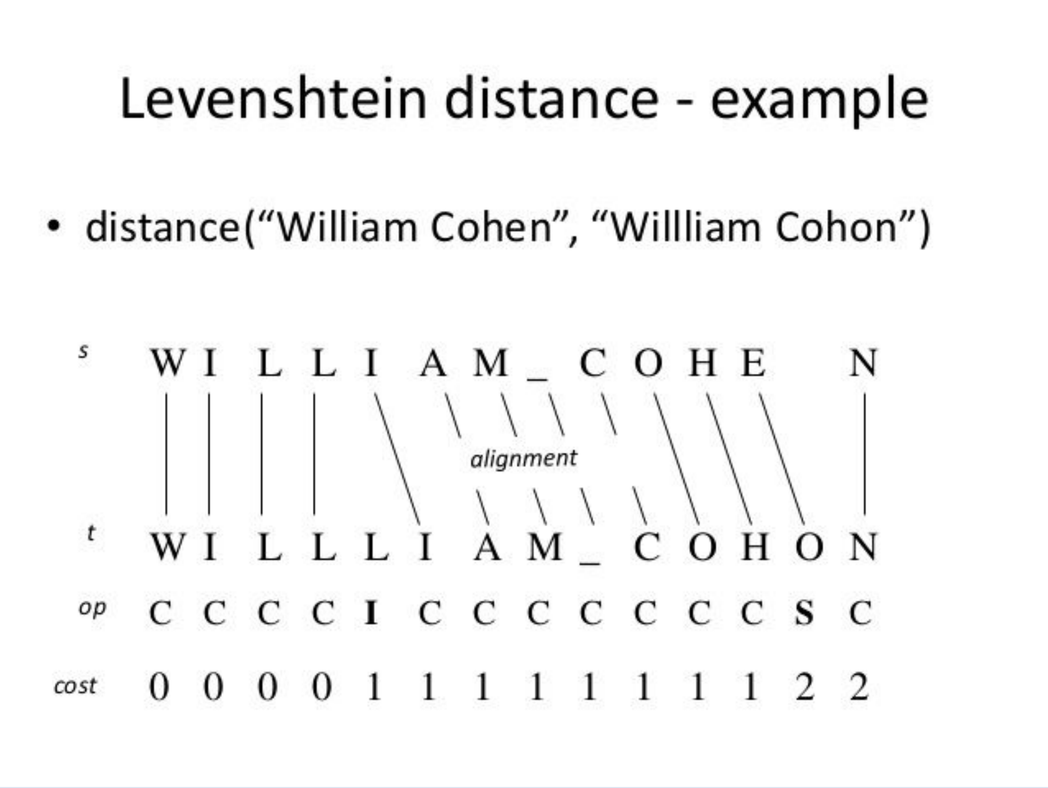
- Automated
  - Exact Matching (True if everything matches)
  - Rule Based Matching
  - Score Based Matching : String Comparators

Data Linkage basics (in brief)

- Need way to find way of quantifying similarity / distance between 2 strings (feature engineering)

- String comparators : Character based : Levenshtein edit distance

    - Operations
        - **I**nsertion
        - **S**ubstitution
        - **D**eletion
        - **T**ransposition *
        - **C**opy (no change)



Data Linkage basics (in brief)

- Need way to find way of quantifying similarity / distance between 2 strings (feature engineering)

- String comparators : Character based : Jaro & Jaro Winkler

$$Jaro\ similarity = \begin{cases} 0, \text{ if m=0} \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right), \text{ for m!=0} \end{cases}$$

where:

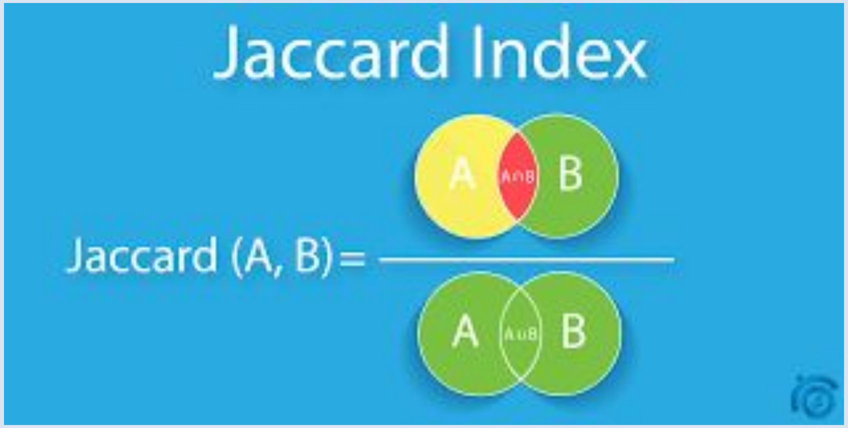
s1,s2 : length of strings to be compared
m : num of matching characters
t : half of num of transpositions

Jaro-Winkler : An extension of Jaro distance. It gives more weight in the agreement of initial prefix characters

Data Linkage basics (in brief)

- Need way to find way of quantifying similarity / distance between 2 strings (feature engineering)
- String comparators: sub-string (Qgram) based : Jaccard Similarity

John   = {“Jo”,”oh”,”hn”)
Johnny = {“Jo”,”oh”,”hn”,”nn”,”ny”)
Andrew = {“An”,”nd”,”dr”,”re”,”ew}
Andy   = {“An”,”nd”,”dy”)

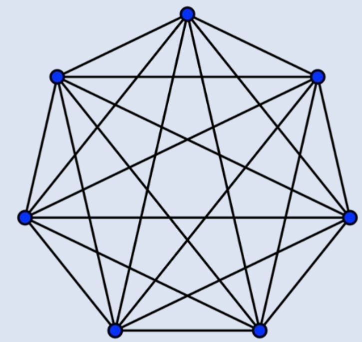


$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Blocking (in brief)

- **Candidate Pairs**
    - Problem: too many comparisons
    - **O(N²)**

      10 rows ~ 100 comparisons 🤷‍♂️🤷‍♀️
    - 1mil rows ~ 1 tril comparisons 💣



- **Blocking**:

  Divide datasets into groups, called blocks, in order to ***reduce*** **the comparison** ***space*** **to only those matched pairs that meet certain basic criteria.** Only records in corresponding blocks on each dataset are compared, to identify possible links.

- "We don't mind comparing apples with oranges. We just don't want to compare apples with toasters" - from Sam Lindsay's pretend TED talk on blocking 🙃

Blocking (in brief)

- **Blocking on Month of Birth Example**

Blocking (in brief)

- **Double Metaphone Phonetic Encoding helps having more permissive blocking rules**

[LNKR]

Lineker

Linacre

Fellegi-Sunter theory (in brief)

- Fellegi and Sunter (1969) devised a way of translating the problem of record linkage into a statistical model

- This gives us a functional form, parameters to estimate, and a likelihood function

Model input:	Two records to be compared
Model output:	The probability that these records are a match

- Each linking variable considered separately and assumed independent of the others

    - If we see that first name matches, how much new information does this give us?

    - If we see that first name does not match, how much new information does this give us?

Fellegi-Sunter theory (in brief)

For example:

Record	Forename	Surname	Sex	NI number
A	John	Smith	M	AB12345C
B	Jonathan	Smith	F	AB12345C
similarity vector	**1**	**2**	**0**	**2**

No match = 0
Partial match = 1
Exact match = 2

1) *What is the probability of each of these 4 outcomes…*
   - *…if A and B are the same person?*
   - *…if A and B are not the same person?*

2) *How important are each of these outcomes in determining a link?*

Fellegi-Sunter theory (in brief)

For example:

Record	Forename	Surname	Sex	NI number
A	John	Smith	M	AB12345C
B	Jonathan	Smith	F	AB12345C
similarity vector	1	2	0	2

No match = 0
Partial match = 1
Exact match = 2

1) *What is the probability of each of these 4 outcomes…*
   - *…if A and B are the same person?* **m probability**
   - *…if A and B are not the same person?* **u probability**

Sex similarity level	**0**	**1**	**2**
m (match)	0.05	0	0.95
u (unmatch)	0.5	0	0.5

Probability of a match
= m / (m+u)
= 0.05 / 0.505
= 0.1

Fellegi-Sunter theory (in brief)

For example:

Record	Forename	Surname	Sex	NI number
A	John	Smith	M	AB12345C
B	Jonathan	Smith	F	AB12345C
similarity vector	1	2	0	2

No match = 0
Partial match = 1
Exact match = 2

1) *What is the probability of each of these 4 outcomes…*
   - *…if A and B are the same person?* **m probability**
   - *…if A and B are not the same person?* **u probability**

NI number similarity level	**0**	**1**	**2**
m (match)	0.002	0.008	0.99
u (unmatch)	0.998	0.0015	0.0005

Probability of a match
= m / (m+u)
= 0.99 / 0.9905
= 0.9995

Fellegi-Sunter theory (in brief)

For example:

Record	Forename	Surname	Sex	NI number
A	John	Smith	M	AB12345C
B	Jonathan	Smith	F	AB12345C
similarity vector	1	2	0	2
Bayes Factor K = m / u			1/10	1980

No match = 0
Partial match = 1
Exact match = 2

2) *How important are each of these outcomes in determining a link?*

Bayes Factor, K: ratio of m and u probabilities for a given similarity level

Using the example m and u probablities for Sex and NI number:

- It is **10x** more likely that A and B **do not match**, given Sex does not match
- It is almost **2000x** more likely that A and B **match**, given NI number matches

→ NI number has more influence than Sex in determining a link

Aims of splink:

- Work at much **greater scale** than current open source implementations (>100 million records).
- Get results **faster** than current open source implementations - with runtimes of less than an hour even for large record linking problems.
- Have a highly **transparent methodology**, so the match scores can be easily explained both graphically and in words
- Have **accuracy** similar to some of the best alternatives, open source or commercial
- Give linked data users **access to link confidence** so they can perform sensitivity analysis
- Considerable **flexibility** and customizability enables it to tackle the majority of record linking and deduplication problems
- **Robust**. Automated suite of quality assurance tests.

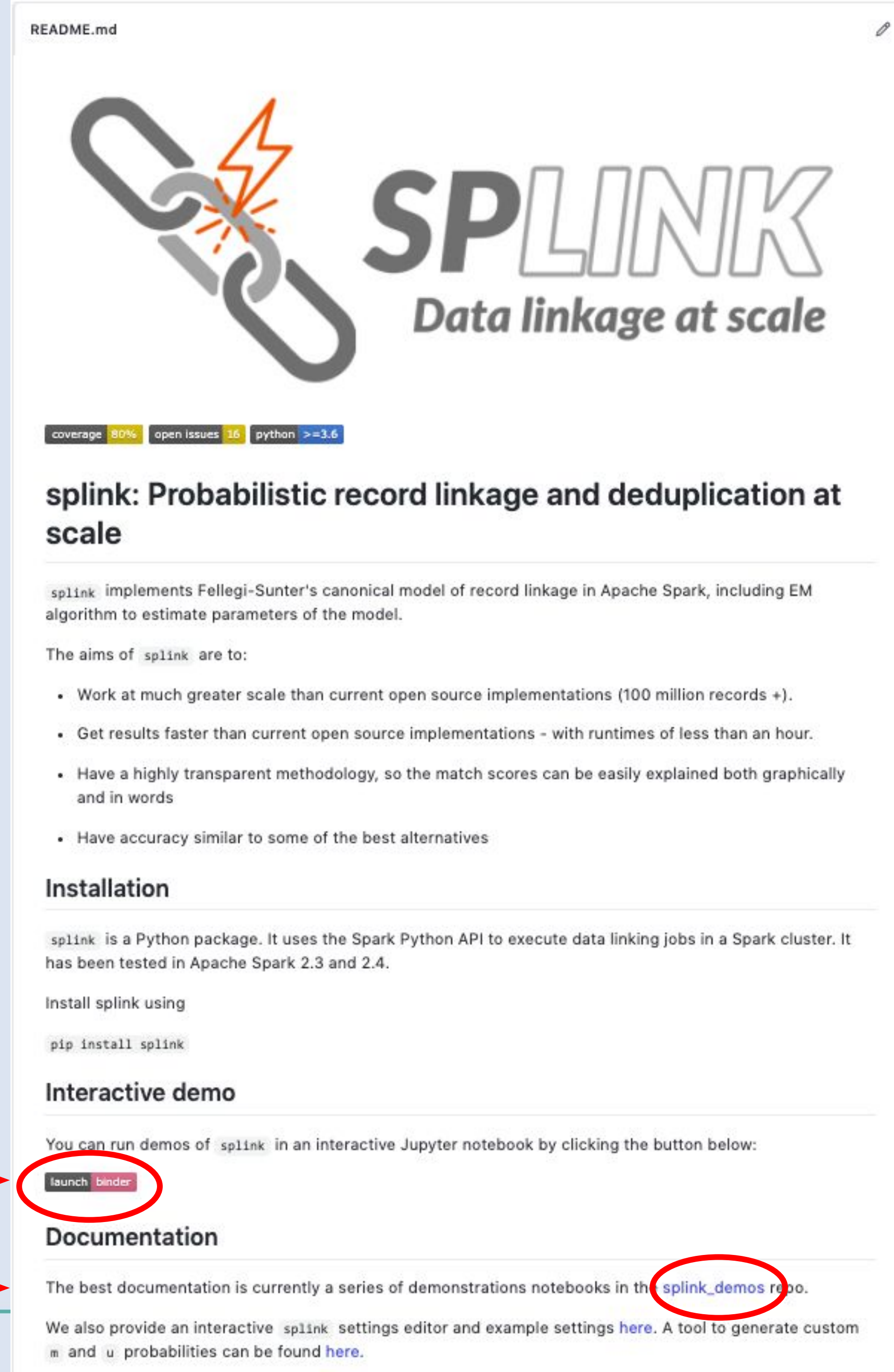
Demo

The **splink** code can be found at:

github.com/moj-analytical-services/splink

You can run an interactive demo of **splink** against a real (tiny!) Apache Spark server using notebooks published here:

github.com/moj-analytical-services/splink_demos



README.md

SPLINK
Data linkage at scale

coverage 80% | open issues 16 | python >=3.6

splink: Probabilistic record linkage and deduplication at scale

`splink` implements Fellegi-Sunter's canonical model of record linkage in Apache Spark, including EM algorithm to estimate parameters of the model.

The aims of `splink` are to:

- Work at much greater scale than current open source implementations (100 million records +).
- Get results faster than current open source implementations - with runtimes of less than an hour.
- Have a highly transparent methodology, so the match scores can be easily explained both graphically and in words
- Have accuracy similar to some of the best alternatives

Installation

`splink` is a Python package. It uses the Spark Python API to execute data linking jobs in a Spark cluster. It has been tested in Apache Spark 2.3 and 2.4.

Install splink using

`pip install splink`

Interactive demo

You can run demos of `splink` in an interactive Jupyter notebook by clicking the button below:

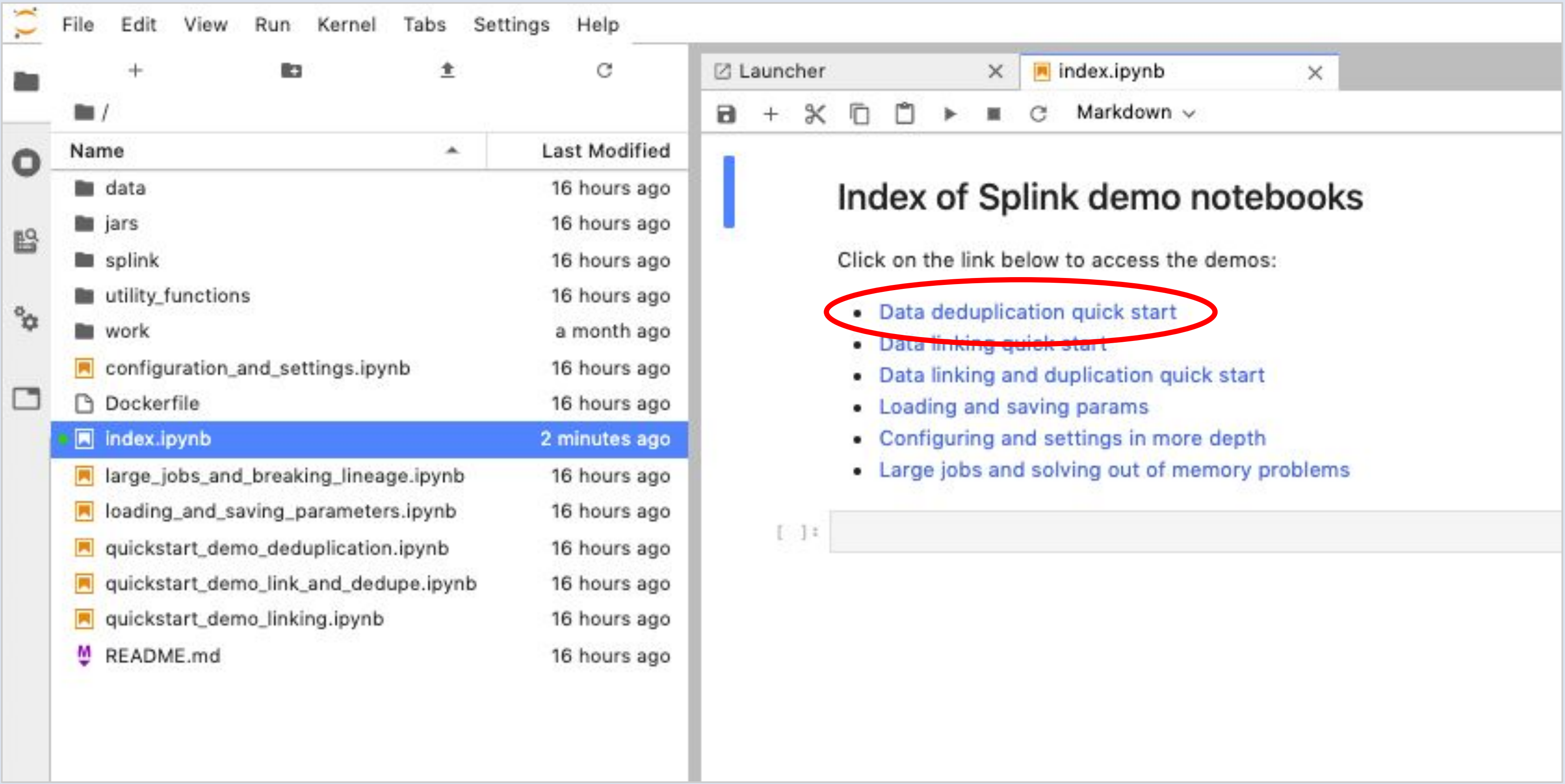
launch binder

Documentation

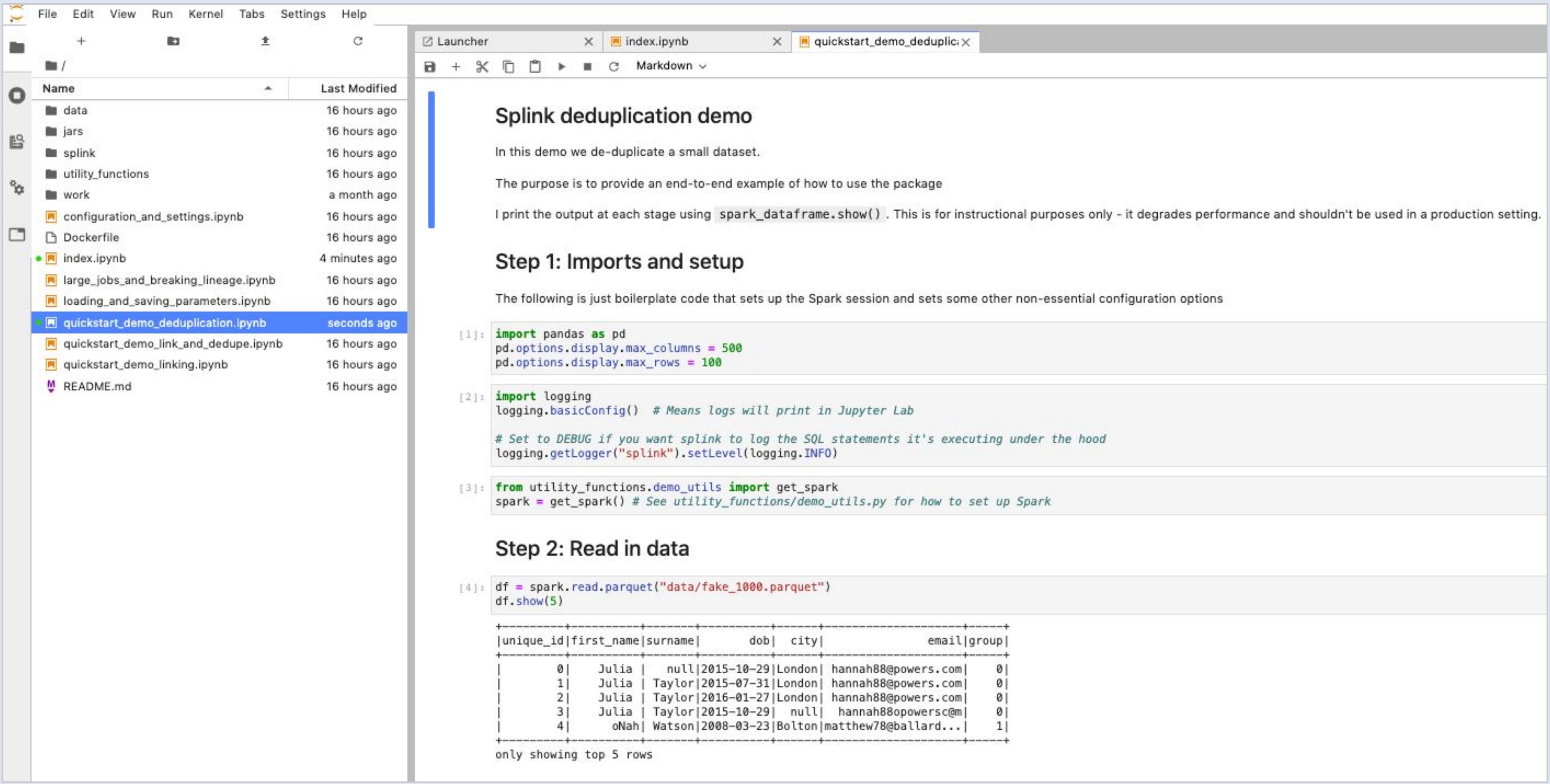
The best documentation is currently a series of demonstrations notebooks in the splink_demos repo.

We also provide an interactive `splink` settings editor and example settings here. A tool to generate custom `m` and `u` probabilities can be found here.

Demo


File Edit View Run Kernel Tabs Settings Help
Name
Last Modified
data 16 hours ago
jars 16 hours ago
splink 16 hours ago
utility_functions 16 hours ago
work a month ago
configuration_and_settings.ipynb 16 hours ago
Dockerfile 16 hours ago
index.ipynb 2 minutes ago
large_jobs_and_breaking_lineage.ipynb 16 hours ago
loading_and_saving_parameters.ipynb 16 hours ago
quickstart_demo_deduplication.ipynb 16 hours ago
quickstart_demo_link_and_dedupe.ipynb 16 hours ago
quickstart_demo_linking.ipynb 16 hours ago
README.md 16 hours ago
Launcher
index.ipynb
Markdown
Index of Splink demo notebooks
Click on the link below to access the demos:
Data deduplication quick start
Data linking quick start
Data linking and duplication quick start
Loading and saving params
Configuring and settings in more depth
Large jobs and solving out of memory problems

Deduplication of fake data



Deduplication of fake data

Step 2: Read in data

```
[4]: df = spark.read.parquet("data/fake_1000.parquet")
df.show(5)
```

```
+---------+----------+-------+----------+------+--------------------+-----+
|unique_id|first_name|surname|       dob|  city|               email|group|
+---------+----------+-------+----------+------+--------------------+-----+
|        0|    Julia |   null|2015-10-29|London| hannah88@powers.com|    0|
|        1|    Julia | Taylor|2015-07-31|London| hannah88@powers.com|    0|
|        2|    Julia | Taylor|2016-01-27|London| hannah88@powers.com|    0|
|        3|    Julia | Taylor|2015-10-29|  null|  hannah88opowersc@m|    0|
|        4|      oNah| Watson|2008-03-23|Bolton|matthew78@ballard...|    1|
+---------+----------+-------+----------+------+--------------------+-----+
only showing top 5 rows
```

Deduplication of fake data

Step 3: Configure splink using the `settings` object

```
[5]: settings = {
    "link_type": "dedupe_only",
    "blocking_rules": [
        "l.first_name = r.first_name",
        "l.surname = r.surname",
        "l.dob = r.dob"
    ],
    "comparison_columns": [
        {
            "col_name": "first_name",
            "num_levels": 3,
            "term_frequency_adjustments": True
        },
        {
            "col_name": "surname",
            "num_levels": 3,
            "term_frequency_adjustments": True
        },
        {
            "col_name": "dob"
        },
        {
            "col_name": "city"
        },
        {
            "col_name": "email"
        }
    ],
    "additional_columns_to_retain": ["group"],
    "em_convergence": 0.01
}
```

Most **splink** configuration options are stored in a settings dictionary.

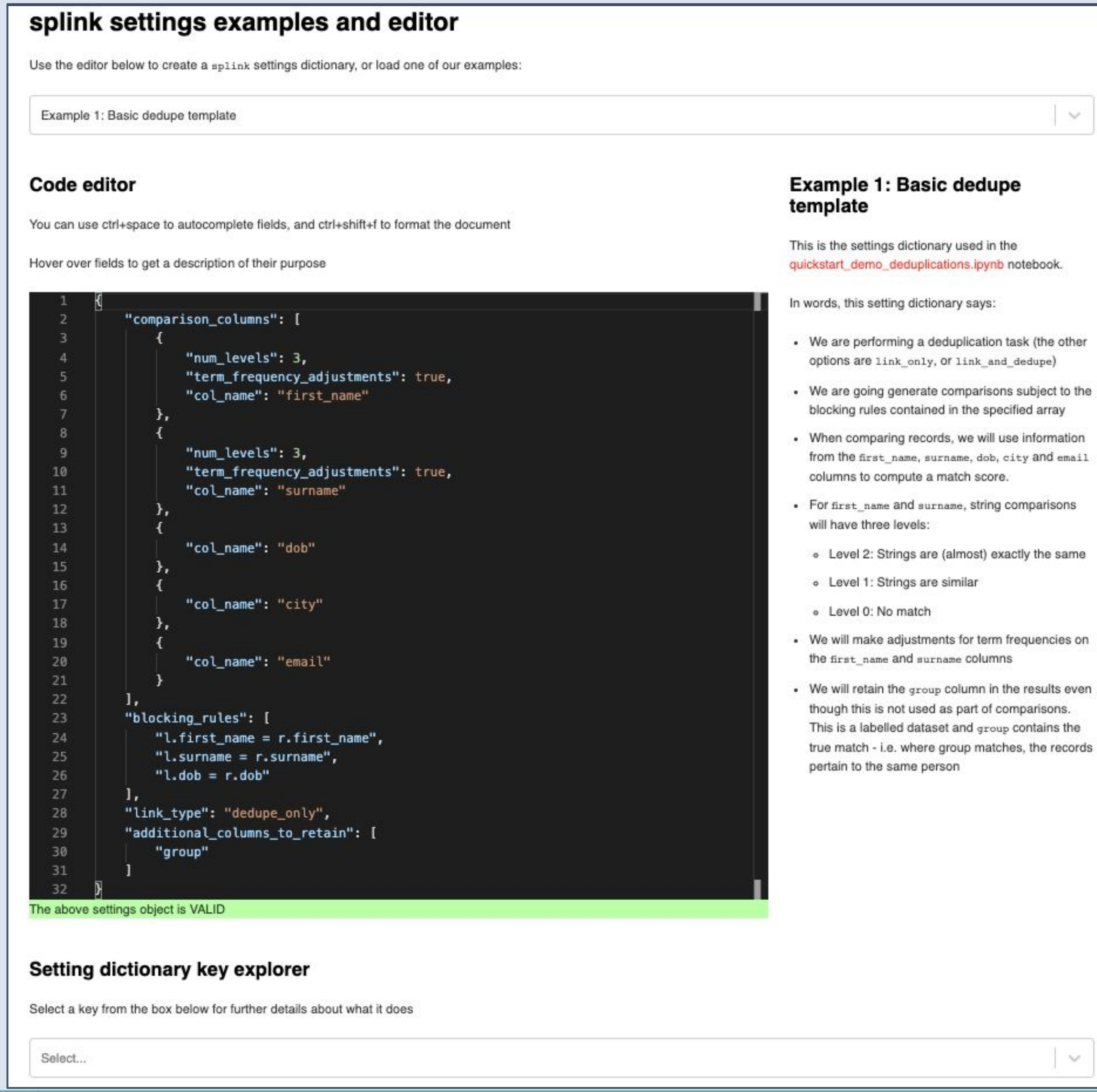
This dictionary allows significant customisation, and can therefore get quite complex.

Settings editor

We provide an online tool for helping to write valid settings dictionaries, which includes:

- Tooltips
- Autocomplete
- Examples for various scenarios
- Documentation for all available settings dictionary keys

moj-analytical-services.github.io/splink_settings_editor/



Deduplication of fake data

Step 4: Estimate match scores using the Expectation Maximisation algorithm

Columns are assumed to be strings by default. See the 'comparison vector settings' notebook for details of configuration options.

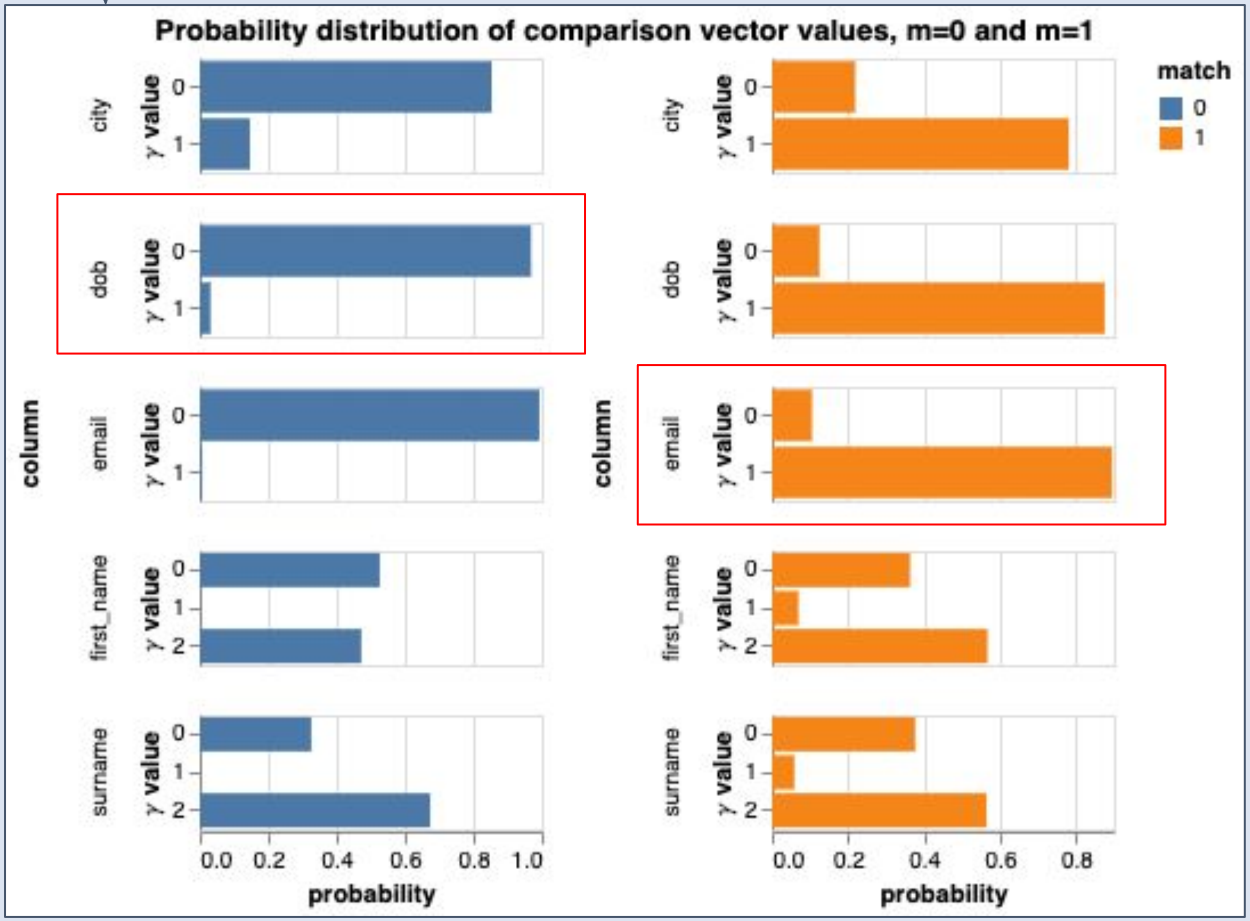
[6]:
```python
from splink import Splink

linker = Splink(settings, spark, df=df)
df_e = linker.get_scored_comparisons()
```

```
INFO:splink.iterate:Iteration 0 complete
INFO:splink.params:The maximum change in parameters was 0.5087412834167481 for key π_gamma_surname_prob_dist_non_match_level_2_probability
INFO:splink.iterate:Iteration 1 complete
INFO:splink.params:The maximum change in parameters was 0.0954439640045166 for key π_gamma_surname_prob_dist_match_level_2_probability
INFO:splink.iterate:Iteration 2 complete
INFO:splink.params:The maximum change in parameters was 0.021286725997924805 for key π_gamma_dob_prob_dist_non_match_level_0_probability
INFO:splink.iterate:Iteration 3 complete
INFO:splink.params:The maximum change in parameters was 0.010865330696105957 for key π_gamma_dob_prob_dist_non_match_level_0_probability
INFO:splink.iterate:Iteration 4 complete
INFO:splink.params:The maximum change in parameters was 0.008596867322921753 for key π_gamma_email_prob_dist_match_level_0_probability
INFO:splink.iterate:EM algorithm has converged
```

Algorithm runs until parameters converge to a stable solution

Deduplication of fake data

```
params = linker.params
params.probability_distribution_chart()
```



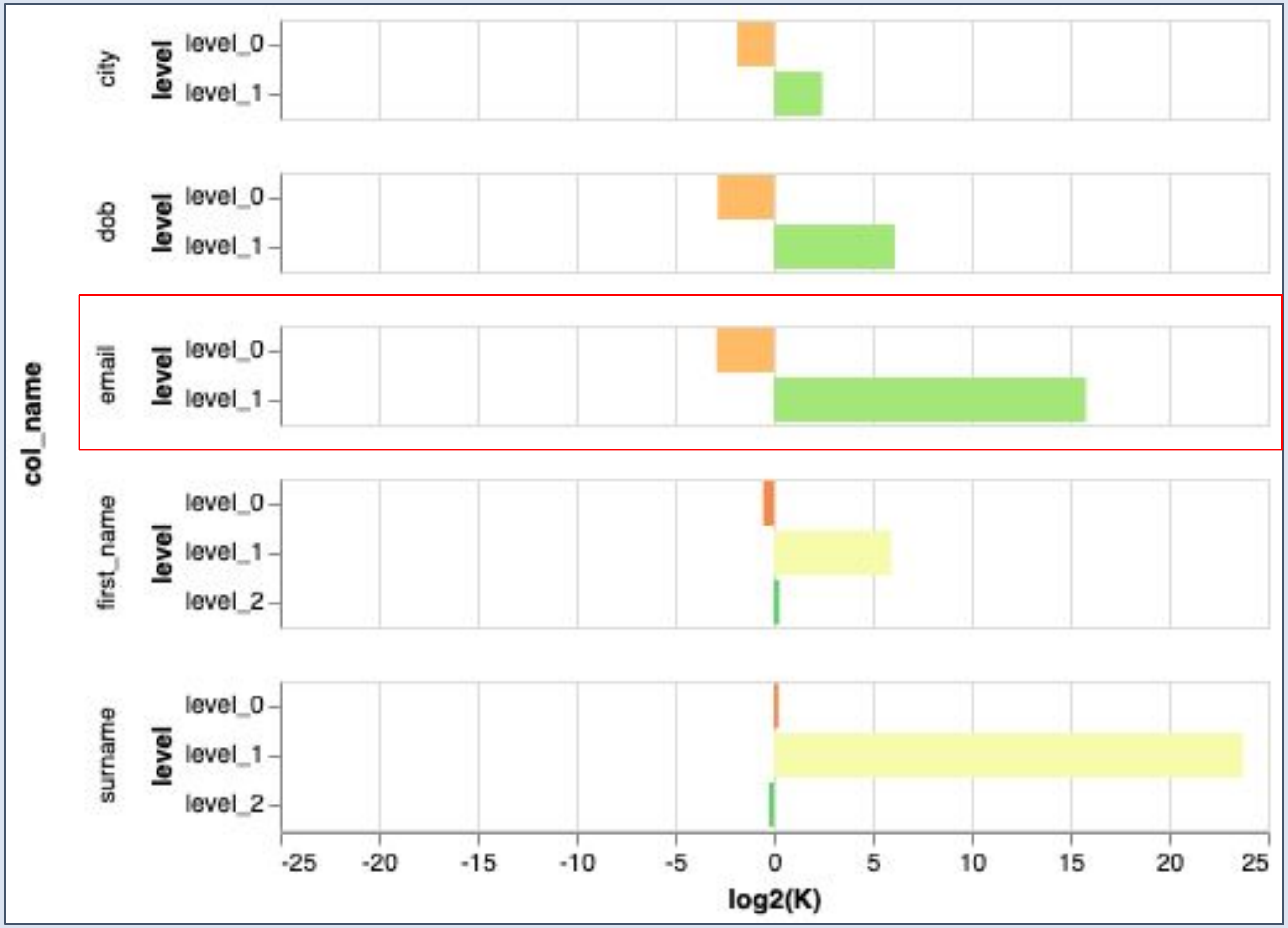
Translation:
Among non-matches probability of DOB matching is <5%

Translation:
Among matches probability of email matching is ~90%

u probabilities

m probabilities

Deduplication of fake data



Translation:
A *positive* match on e-mail is a very strong indicator of a match (2^15 times more likely than not), but a *negative* match on e-mail is not as strong an indicator of a non-match

log(Bayes Factor)

Deduplication of fake data

Step 5: Inspect results

[7]:
```
# Inspect main dataframe that contains the match scores
cols_to_inspect = ["match_probability","unique_id_l","unique_id_r","group_l", "group_r", "first_name_l","first_name_r","surname_l","surname_r","dob_l","dob_r","city_l","city_r","email_l

df_e.toPandas()[cols_to_inspect].sort_values(["unique_id_l", "unique_id_r"]).head(10)
```

[7]:

	match_probability	unique_id_l	unique_id_r	group_l	group_r	first_name_l	first_name_r	surname_l	surname_r	dob_l	dob_r	city_l	city_r	email_l	email_r
2	0.985811	0	1	0	0	Julia	Julia	None	Taylor	2015-10-29	2015-07-31	London	London	hannah88@powers.com	hannah88@powers.com
1	0.985811	0	2	0	0	Julia	Julia	None	Taylor	2015-10-29	2016-01-27	London	London	hannah88@powers.com	hannah88@powers.com
0	0.999646	0	3	0	0	Julia	Julia	None	Taylor	2015-10-29	2015-10-29	London	None	hannah88@powers.com	hannah88opowersc@m
4	0.983115	1	2	0	0	Julia	Julia	Taylor	Taylor	2015-07-31	2016-01-27	London	London	hannah88@powers.com	hannah88@powers.com
3	0.916171	1	3	0	0	Julia	Julia	Taylor	Taylor	2015-07-31	2015-10-29	London	None	hannah88@powers.com	hannah88opowersc@m
2290	0.027342	1	89	0	18	Julia	Chirla	Taylor	Taylor	2015-07-31	2006-06-28	London	London	hannah88@powers.com	mbrooks@booker.com
2289	0.027342	1	142	0	26	Julia	Harry	Taylor	Taylor	2015-07-31	2017-11-24	London	London	hannah88@powers.com	coltonray@lee.com
2288	0.027342	1	148	0	26	Julia	Harry	Taylor	Taylor	2015-07-31	2017-09-01	London	London	hannah88@powers.com	coltonray@lee.com
4821	0.792436	1	246	0	43	Julia	Harrison	Taylor	Joshua	2015-07-31	2015-07-31	London	Southend-on-Sea	hannah88@powers.com	None
2287	0.039123	1	362	0	62	Julia	None	Taylor	Taylor	2015-07-31	1989-07-25	London	London	hannah88@powers.com	wagnershane@landry.com

✓ All comparisons between IDs 0, 1, 2 and 3 (Julia Taylor) have a match probability > 0.9

Next steps (beyond the scope of this presentation):

- Match threshold - what score constitutes a confirmed match?
- Transitive links & resolving the graph
- QA - making sure the results agree with clerical matching

Some DOs and DON'Ts

- Make sure you have unique row IDs for your input data
- Clean/standardise your data before linking
- Avoid linking on highly correlated fields (e.g. postcode and street)
- Please get in contact on #data_linkage_deduplication in D&T Slack (mojdt)

General guidance on best practices for using **splink**, with more detailed explanation and suggestions can be found in the splink_demos repo:

https://github.com/moj-analytical-services/splink_demos/blob/master/best_practices.ipynb