FINDING POLITICAL FACTS & RELATIONS WITH WIKIDATA

Theresa Gessler

TODAY

- Introduction & Wikipedia
- Wikidata
- Six degrees of separation
- Tracing legacies of slave-ownership with Wikidata
- Hands-on: Querying Wikidata

INTRODUCTION & WIKIPEDIA

INTRODUCTION

- Comparative Politics @ Europa-Universität Viadrina
  - democracy
  - immigration
  - gender
  - using text & digital data
- theresagessler.eu
  - @gessler@fediscience.org
  - @th_ges

Wikipedia monument in Słubice

INTRODUCTION: RESEARCH ON WIKIPEDIA

- Wikipedia as large online encyclopedia ↔ Wikipedia as a data source

- 3 types of questions
  - articles & networks: representation & bias
  - pageviews: interest in topics
  - clickstreams: how do users navigate information?

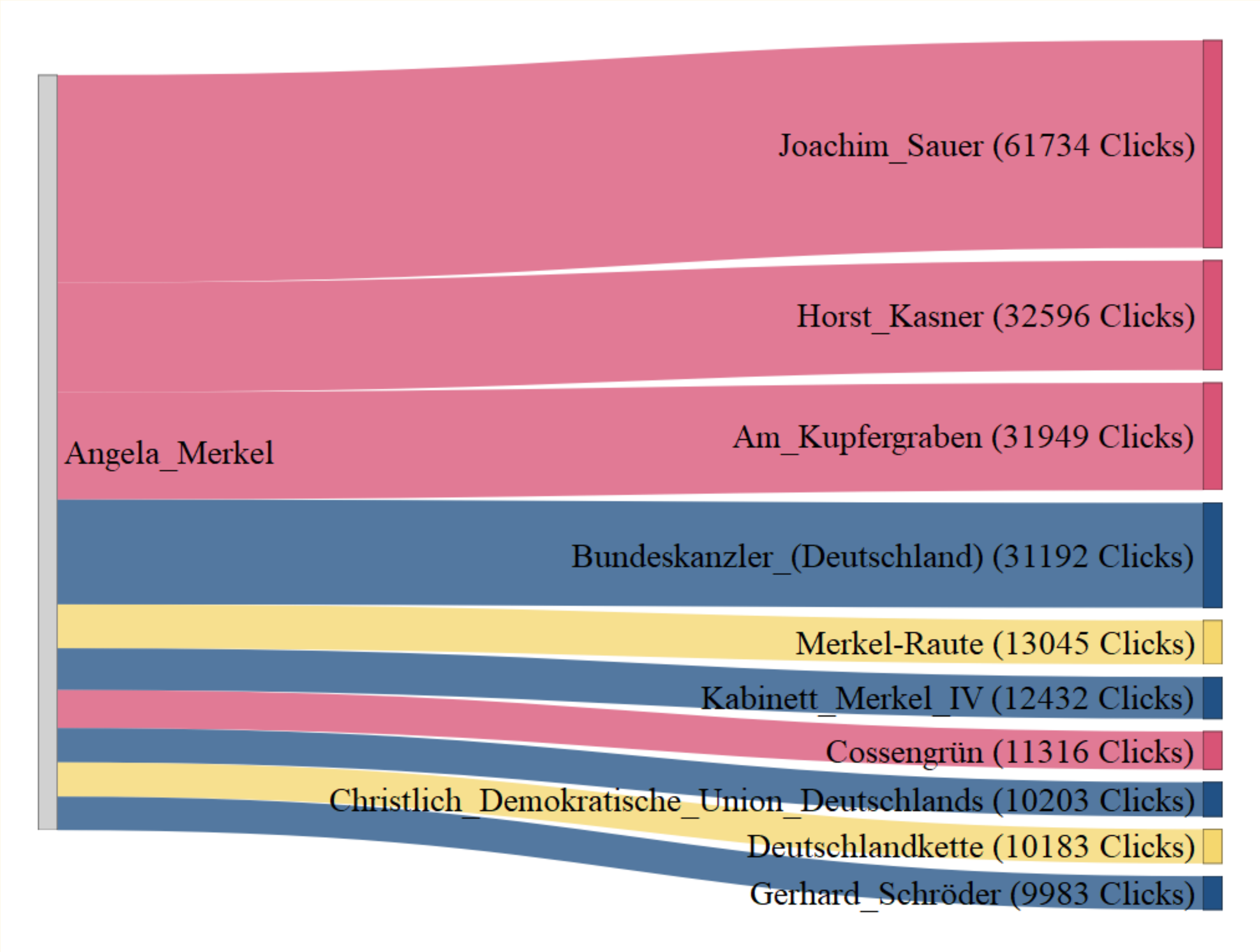
- → Wikipedia as a place where politics becomes visible
  - differences to social media & other online data sources

INTRODUCTION: RESEARCH ON WIKIPEDIA

- Using Wikipedia to measure political phenomena - e.g. gender bias
  - bias in Wikipedia content (Pradel 2020, Wagner 2015, Wagner 2016)
  - bias in networks (Langrock & González-Bailón 2022)
  - **bias in user behavior?**
- using links on Wikipedia pages & matching them to dyadic clickstream data
  - how often do users click from article X to article Y per month?
- classification of link content
  - analysis of clicks for links of certain types
  - direct & interaction effects of politicians' gender on users' interest

INTRODUCTION: RESEARCH ON WIKIPEDIA

EXAMPLE: ANGELA MERKEL



MZES SSDL RESSOURCES

- Studying politics on and with Wikipedia @ MZES SSDL (Denis Cohen, Nick Baumann, Simon Munzert)
  - pageviews
  - article links
  - clickstream data
  - Wikidata
    - legislatoR

→ encompassing intro to using Wikipedia for political scientists

WIKIDATA

WIKIDATA

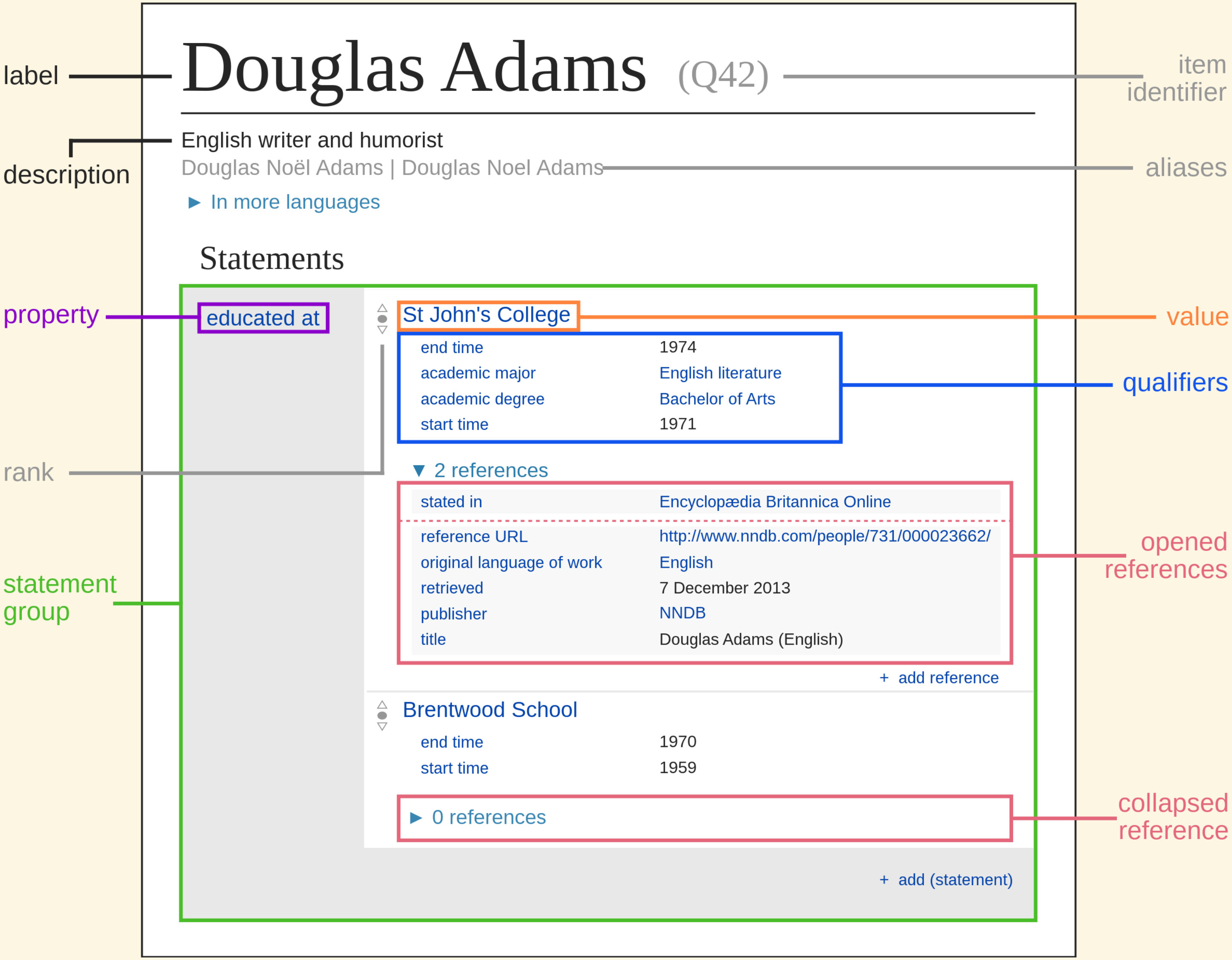
- open knowledge database
  - public domain → reuse even for commercial purposes
  - collaborative: everyone can edit
  - structured 'statements' & secondary database → machine readable & linked to other databases
- used in Wikipedia & other projects

- example: CDU
- example: Philipp Amthor

TERMINOLOGY

- each Wikidata item has an **ID** (starting with Q - e.g.: “Q64032638”)
    - for human readers: **label** and **description** (not unique)
- **statements** provide information
    - Wikidata items have **properties** (starting with P - e.g.: “P39”)
    - properties of an item have **values**: other Wikidata items (e.g. “Q27169”)

TERMINOLOGY



Douglas Adams (Q42)

English writer and humorist

Douglas Noël Adams | Douglas Noel Adams

► In more languages

Statements

educated at

St John's College

end time	1974
academic major	English literature
academic degree	Bachelor of Arts
start time	1971

▼ 2 references

stated in	Encyclopædia Britannica Online
reference URL	http://www.nndb.com/people/731/000023662/
original language of work	English
retrieved	7 December 2013
publisher	NNDB
title	Douglas Adams (English)

+ add reference

Brentwood School

end time	1970
start time	1959

► 0 references

+ add (statement)

WIKIDATA VS. WIKIPEDIA

WIKIPEDIA

- continuous text → focused on human readers
- ~ 6 million items (english)
- standards for inclusion: notability, substance, verifiability, someone who edits

WIKIDATA

- machine readable / linked data
- 100 million items (multilingual)
- standards for inclusion: verifiability, someone who edits
  - overview: what is in Wikidata

USES

Database for

- **information** about legislators, activists, organizations, constituencies, …
  - stored as properties of entities
- **relations** between entities
  - networks spanned by properties
  - e.g. work by Ömer F. Yalçın: Empirical Study of Elite Networks with Wikidata

EXAMPLE: LEGISLATOR

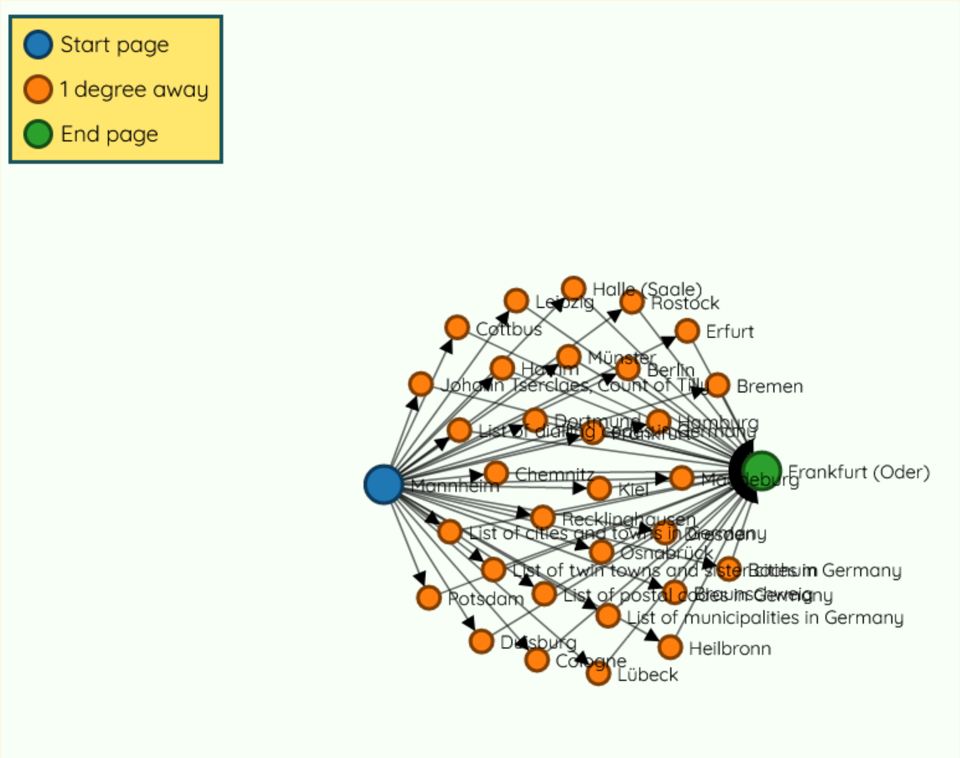
- R package by Sascha Göbel & Simon Munzert
- Comparative Legislators Database (CLD)
  - demographic background
  - office & role in party
  - Wikipedia indicators, e.g. traffic
  - identifiers in other datasets

→ key advantage of Wikidata: machine-readable format, only validation to be done by researchers

SIX DEGREES OF SEPARATION

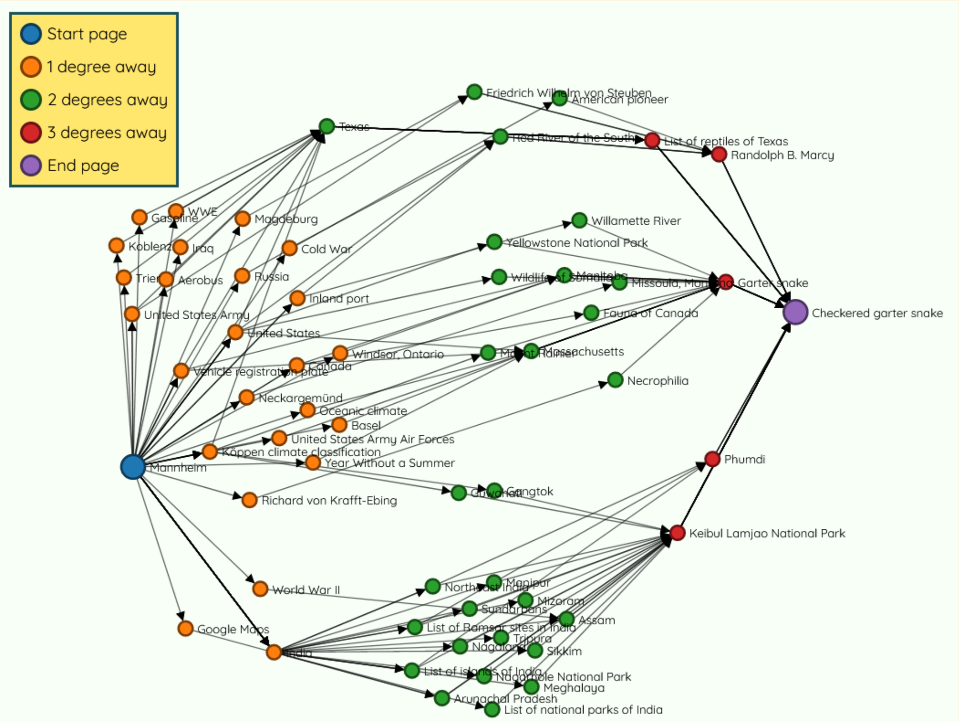
SIX DEGREES OF WIKIPEDIA

- How many steps do you need: Mannheim ↔ Frankfurt (Oder)
  - via Six Degrees of Wikipedia



SIX DEGREES OF WIKIPEDIA

- How many steps do you need: Mannheim ↔ Checkered garter snake
  - via Six Degrees of Wikipedia

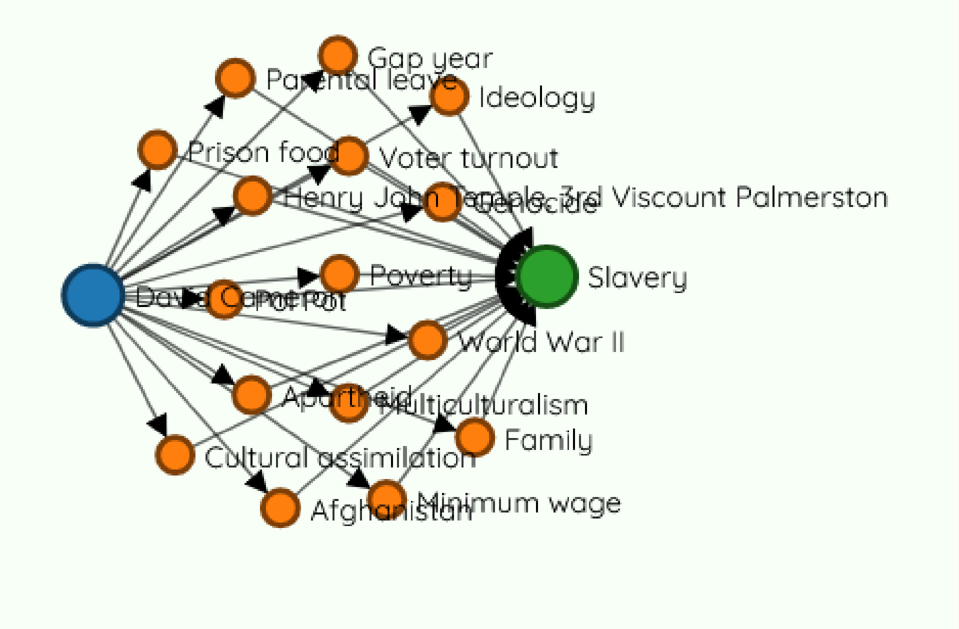


→ Which relations are actually meaningful?

SIX DEGREES OF WIKIPEDIA

→ Which relations are actually meaningful?

- How many steps do you need: David Cameron ↔ Slavery
  - via Six Degrees of Wikipedia



→ We can manually assess a few connections…

→ …but Wikidata helps us to answer 'Which connection is meaningful?' at scale

TRACING LEGACIES OF SLAVE-OWNERSHIP WITH WIKIDATA

Co-authored ongoing work with Joe Kendall (European University Institute)

LEGACIES OF SLAVE-OWNERSHIP

- **legacies** of institutions like slavery have shaped modern societies, including the UK
  - persistence through dynastic & social ties, wealth
- however, **(quantitative) research has been limited**
  - legacies less directly discernible (UK)
  - challenge of quantifying networks
- → show patterns of elite persistence
- → methodological tools for study of social proximity

EMPIRICAL APPROACH

- **collection of Wikidata IDs**
  - slave-owners
  - British MPs
- downloading **statements as potential links** (up to 6 degrees)
  - selection of relevant properties: family ties, business relation, academic relations, sports clubs
  - tracing of paths: slave-owners → MPs in multiple steps
- estimation of **proximity measures** & interesting paths

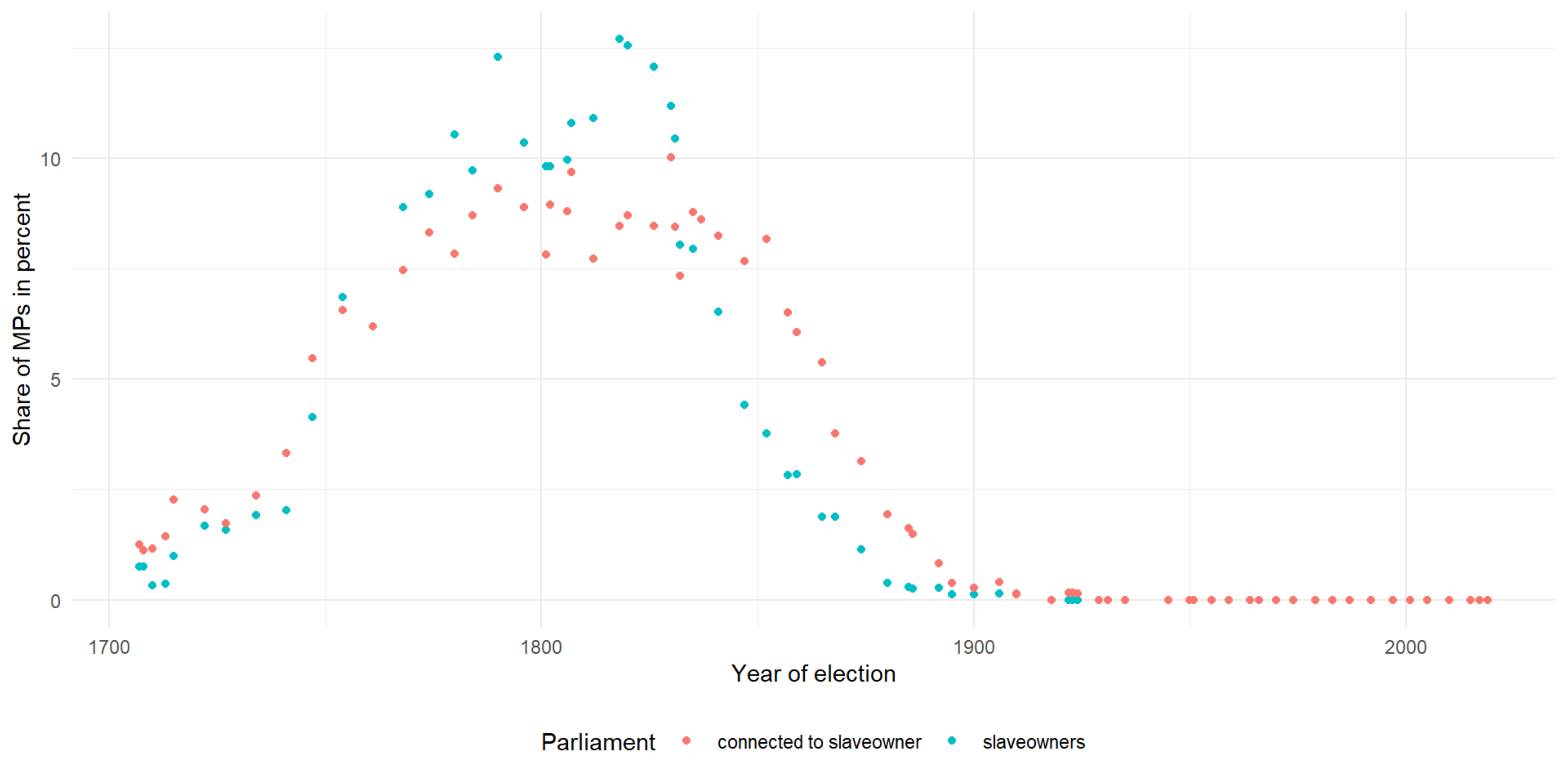
EMPIRICAL APPROACH

- ~ 55.000 MP-terms for 16.000 unique members of parliament
- ~ 1600 slave-owners or close relatives from the relevant period in Wikidata
- between 3 and 500 properties per entity → more detailed for current-day entries
- recurring dyads
- rapid growth of network
- decreasing return of connections (3-14%)

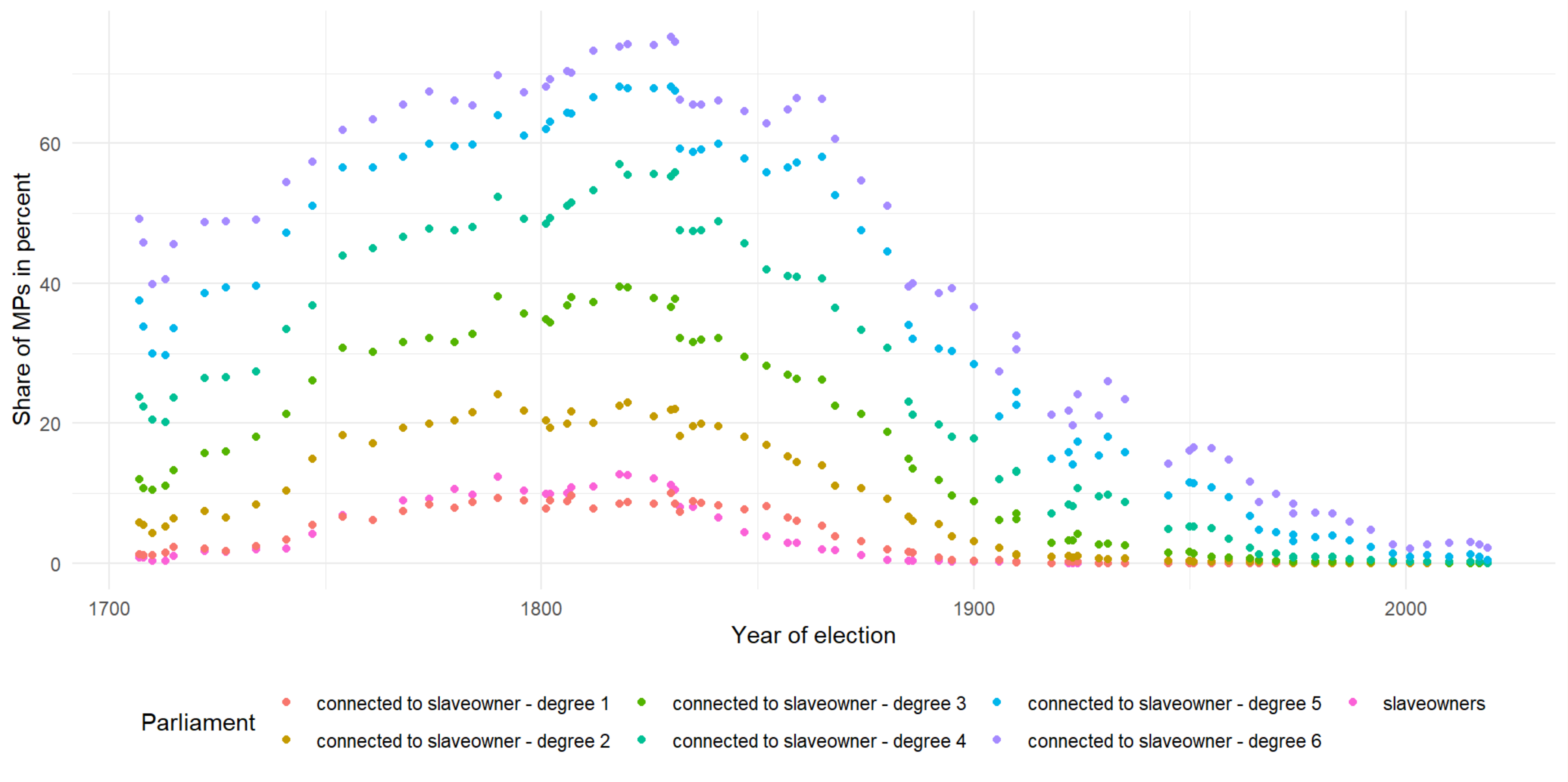
FREQUENT PROPERTIES - DYADS

```
# A tibble: 20 × 3
   label                  property      n
   <chr>                  <chr>     <int>
 1 child                  P40      252874
 2 father                 P22      116012
 3 spouse                 P26      106814
 4 mother                 P25       99457
 5 sibling                P3373     35720
 6 noble title            P97        7722
 7 military branch        P241       4229
 8 student of             P802       2119
 9 relative               P1038      2043
10 noble family           P53        1936
11 family                 P53        1936
12 student of             P1066      1345
13 owner of               P1830      1152
14 doctoral student       P185       1047
15 influenced by          P737        657
16 cohabitant             P451        473
17 member of sports team  P54         428
18 doctoral advisor       P184        377
19 godparent              P1290       284
```

SOME PRELIMINARY RESULTS



SOME PRELIMINARY RESULTS



CHALLENGES

- Wikidata **only accumulates data that already exists**
  - potential challenge for historical research
  - privileged status of family relations
- **data quality** depends on the subject
  - coverage as main issue
  - notability bias, also on Wikidata
- **conceptual challenges**
  - assymmetries in certain links, e.g.: political party members, doctoral advisers, …
  - for historical research: chronology

WRAP-UP: WIKIDATA

- open, accessible & growing database of political facts
  - interlinked with other sources
  - multilingual & crowdsourced
- key advantages
  - database of statements about entities
  - tracing networks, including over several connections
  - qualifying connections by properties

HANDS-ON: QUERYING WIKIDATA

WIKIDATA QUERY SERVICE

- uses SPARQL (a query language for databases) on Wikidata Query Service
  - can be queried via `WikidataQueryServiceR`

SPARQL query example:

```
SELECT ?item  ?itemLabel ?itemDescription
WHERE
{
  ?item wdt:P39 wd:Q27169.
SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
```

More examples: Wikidata:WikiProject British Politicians Example Queries

WIKIDATA QUERY SERVICE WITH `tidywikidatar`

`tidywikidatar` package provides tidy data & allows caching

→ easier analysis for most political scientists

```r
library(tidywikidatar)
library(dplyr)
```

theresagessler.eu/wikidata.r

…But for larger data collection, WikidataR or WikidataQueryServiceR are faster options

EXAMPLE: MEMBERS OF THE EUROPEAN PARLIAMENT

- build a query with `tw_query()`
- property: P39, position held (public office)
- value: Q27169, member of the European parliament

```r
mep_query <- tw_query(query = c(p = "P39", q = "Q27169"))
head(mep_query)
```

```
# A tibble: 6 × 3
  id    label                        description
  <chr> <chr>                        <chr>
1 Q157  François Hollande            French official and statesman
2 Q329  Nicolas Sarkozy              President of France from 2007 to 2012
3 Q1220 Giorgio Napolitano           11th President of Italy
4 Q1275 Gladwyn Jebb                 acting Secretary-General of the United
Nations…
5 Q2105 Jacques Chirac               President of France from 1995 to 2007
6 Q2124 Valéry Giscard d'Estaing French official and statesman (1926–2020)
```

EXAMPLE: MEMBERS OF THE EUROPEAN PARLIAMENT

```
WikidataQueryServiceR::query_wikidata('SELECT ?item  ?itemLabel ?itemDescri
WHERE
{
  ?item wdt:P39 wd:Q27169.
SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
')
```

EXAMPLE: COMPLEX QUERY

```
# parliamentary terms of a single person
# here, Winston Churchill
SELECT DISTINCT ?constituencyLabel ?partyLabel ?start ?electionLabel ?end ?causeLabel {
 wd:Q8016 p:P39 ?positionStatement . # all positions held by this person
  ?positionStatement ps:P39 [wdt:P279* wd:Q16707842] . # filter to positions which are a subclass of UK MP
 OPTIONAL { ?positionStatement pq:P768 ?constituency . }  # then find various specific values for each term
 OPTIONAL { ?positionStatement pq:P4100 ?party . }
 OPTIONAL { ?positionStatement pq:P580 ?start . }
 OPTIONAL { ?positionStatement pq:P2715 ?election . }
 OPTIONAL { ?positionStatement pq:P582 ?end . }
 OPTIONAL { ?positionStatement pq:P1534 ?cause . }
 SERVICE wikibase:label { bd:serviceParam wikibase:language 'en' }
}
ORDER BY ?start
```

Example from WikiProject British Politicians

STARTING FROM WIKIPEDIA: CURRENT MEMBERS FROM GERMANY

We can also **start from Wikipedia pages** - for example the List of members of the European Parliament for Germany, 2019–2024

```r
mep_de_df <- tw_get_wikipedia_page_links(url = "https://en.wikipedia.org/wiki/List_of_members_of_the_Europear
```

```r
# columns in dataset
colnames(mep_de_df)
```

```
[1] "source_title_url"       "source_wikipedia_title" "source_qid"
[4] "wikipedia_title"        "wikipedia_id"           "qid"
[7] "description"            "language"
```

```r
# titles
sample(mep_de_df$wikipedia_title,10)
```

```
 [1] "European Conservatives and Reformists"
 [2] "Moritz Körner"
 [3] "List of members of the European Parliament for Luxembourg, 1999–2004"
 [4] "List of members of the European Parliament for Sweden, 2019–2024"
 [5] "List of members of the European Parliament (1984–1989)"
 [6] "List of observers to the European Parliament for Spain"
 [7] "List of members of the European Parliament for Cyprus, 2014–2019"
 [8] "Sabine Verheyen"
 [9] "List of members of the European Parliament for the Netherlands, 1989–1994"
[10] "List of members of the European Parliament for Hungary, 2014–2019"
```

→ filtering to meaningful entries

CURRENT MEMBERS FROM GERMANY

- filtering with MEP QIDs

```r
# filter to MEPs - combine with previous query
mep_de_df <- mep_de_df %>% filter(qid %in% mep_query$id)

sample(mep_de_df$wikipedia_title,10)
```

```
 [1] "Andreas Schwab"     "Christian Ehler"    "Sven Simon"
 [4] "Birgit Sippel"      "Reinhard Bütikofer" "Udo Bullmann"
 [7] "Petra Kammerevert"  "Bernd Lange"        "Klaus Buchner"
[10] "Markus Buchheit"
```

- or: filtering down by characteristics (human, held office, …)

e.g.

```r
mep_de_df <- mep_de_df %>% pull(qid) %>%
    tw_get_property(p = "P31")  %>% # instance of
    filter(value == "Q5") # human
```

GET PROPERTIES

To learn more about MEPs, we can **collect their properties** using `tw_get()`

```
mep_de_props <- mep_de_df$qid %>%
  tw_get()

mep_de_props
```

```
# A tibble: 3,780 × 4
   id        property value                rank  
   <chr>     <chr>    <chr>                <chr> 
 1 Q64032638 label_en Alexandra Geese      <NA>  
 2 Q64032638 P21      Q6581072             normal
 3 Q64032638 P569     +1968-07-01T00:00:00Z normal
 4 Q64032638 P106     Q333634              normal
 5 Q64032638 P106     Q82955               normal
 6 Q64032638 P31      Q5                   normal
 7 Q64032638 P227     118711913X           normal
 8 Q64032638 P735     Q6081128             normal
 9 Q64032638 P27      Q183                 normal
10 Q64032638 P19      Q586                 normal
# … with 3,770 more rows
```

TYPES OF PROPERTIES

→ `tw_get_property_label()` allows to see labels of frequent properties

```r
properties <- mep_de_props %>%
  group_by(property) %>%
  tally() %>%
  arrange(desc(n)) %>%
  mutate(label=tw_get_property_label(property))
```

TYPES OF PROPERTIES

```r
properties %>% head(20)
```

```
# A tibble: 20 × 3
   property                n label
   <chr>               <int> <chr>
 1 P39                   262 position held
 2 P106                  161 occupation
 3 P8687                 126 social media followers
 4 P937                  122 work location
 5 P102                  114 member of political party
 6 P27                   106 country of citizenship
 7 P735                  103 given name
 8 P569                   98 date of birth
 9 description_en         97 <NA>
10 label_en               97 <NA>
11 P1186                  97 MEP directory ID
12 P1412                  97 languages spoken, written or signed
13 P19                    97 place of birth
```

FILTER: MASTODON IDS

This is typically very up-to date - e.g. Mastodon IDs

```r
mep_de_props %>%
  # filter: mastodon ID property
  filter(property=="P4033")
```

```
# A tibble: 72 × 4
   id        property value                             rank      
   <chr>     <chr>    <chr>                             <chr>     
 1 Q64032638 P4033    alexandra_geese@respublicae.eu deprecated
 2 Q78194    P4033    Andreas_Schwab@respublicae.eu  deprecated
 3 Q74215    P4033    ANiebler@respublicae.eu        deprecated
 4 Q64063467 P4033    anna_cavazzini@respublicae.eu  deprecated
 5 Q16530497 P4033    AxelVossMdEP@respublicae.eu    deprecated
 6 Q65437    P4033    berndlange@respublicae.eu      deprecated
 7 Q108736   P4033    BirgitSippelMEP@respublicae.eu deprecated
 8 Q71660    P4033    ConstanzeKrehl@respublicae.eu  deprecated
 9 Q91526    P4033    ErnstCornelia@respublicae.eu   deprecated
10 Q63532607 P4033    d_boeselager@respublicae.eu    deprecated
# … with 62 more rows
```

Deprecated rank means the data source is known to have errors

MEMBERS OF THE EUROPEAN COUNCIL

→ Try this out e.g. with members of the European Council

MEMBERS OF THE EUROPEAN COUNCIL

Starting from Wikipedia

```r
council_df <- tw_get_wikipedia_page_links(
  url = "https://en.wikipedia.org/wiki/List_of_members_of_the_European_Coun

# filtering to meaningful links
council_members <- council_df %>%
  pull(qid) %>%
  tw_get_property(p = "P31")  %>% # instance of
  filter(value == "Q5") # human
```

OTHER DOMAINS: JUDGES

- e.g.: Q43575168: judge at the Federal Constitutional Court of Germany

```r
judges <- tw_query(query=c(p="P39",q="Q43575168"))

judges_props <-  judges %>%
  pull(id) %>%
  tw_get()

properties <- judges_props %>%
  group_by(property) %>%
  tally() %>%
  arrange(desc(n)) %>%
  mutate(label=tw_get_property_label(property))
```

OTHER DOMAINS: JUDGES

```r
properties %>% head(20)
```

```
# A tibble: 20 × 3
   property           n label
   <chr>          <int> <chr>
 1 P106             230 occupation
 2 P39              198 position held
 3 P166             182 award received
 4 P569             113 date of birth
 5 description_en   109 <NA>
 6 label_en         109 <NA>
 7 P1412            109 languages spoken, written or signed
 8 P19              109 place of birth
 9 P21              109 sex or gender
10 P214             109 VIAF ID
11 P227             109 GND ID
12 P27              109 country of citizenship
13 P31              109 instance of
```

OTHER DOMAINS: JUDGES, SUPREME COURT

- e.g. Q11144: Associate Justice of the Supreme Court of the United States

```
supreme_court <- tw_query(query=c(p="P39",q="Q11144"))


judges_props <-  supreme_court %>%
  pull(id) %>%
  tw_get()

properties <- judges_props %>%
  group_by(property) %>%
  tally() %>%
  arrange(desc(n)) %>%
  mutate(label=tw_get_property_label(property))
```

OTHER DOMAINS: JUDGES, SUPREME COURT

```r
properties %>% head(20)
```

```
# A tibble: 20 × 3
   property             n label
   <chr>            <int> <chr>
 1 P106               315 occupation
 2 P39                295 position held
 3 P69                229 educated at
 4 P735               129 given name
 5 alias_en           120 <NA>
 6 P3430              112 SNAC ARK ID
 7 P18                110 image
 8 P734               109 family name
 9 P569               108 date of birth
10 P102               107 member of political party
11 P27                107 country of citizenship
12 P31                106 instance of
13 description_en     105 <NA>
```

QUERY SERVICE

- while `tidywikidataR` provides a great entry point, it does not include the full spectrum of queries
  - some qualifying information is not included
  - more complex queries (e.g. combinations, qualifiers, …)
  - speed of queries

QUERY SERVICE

- read some introductions
  - ‘gentle introduction’
  - tutorials
  - video introduction for beginners
- play around with examples: Wikidata:WikiProject British Politicians Example Queries
- use the Query Builder

CONCLUSION

- Wikidata as a powerful tool
  - `tidywikidatar` as simple entry point
  - Query Service for more advanced questions & larger datasets
- political facts & relations
  - possibility to qualify relations by types
  - connections to larger social phenomena & institutions (e.g. role of education, sports, …)

THANKS

@gessler@fediscience.org| @th_ges
gessler@europa-uni.de