A New Macro-Micro Approach to the Study of Political Careers

Turner - Zwinkels, Tomas

DOI:
10.33612/diss.131055893

Publication date:
2020

Citation for published version (APA):
Turner - Zwinkels, T. (2020). *A New Macro-Micro Approach to the Study of Political Careers: Theoretical, Methodological and Empirical Challenges and Solutions*. University of Groningen. https://doi.org/10.33612/diss.131055893

3

Chapter 3

PolCa: a Relational Database with Political Career Data

Tomas Turner-Zwinkels

•

TURNER-ZWINKELS, T. (2020). POLCA: A RELATIONAL DATABASE WITH POLITICAL CAREER DATA.

The empirical analyses in the chapters that follow are all based on the same underlying database. In this chapter, I introduce this so-called 'PolCa' database. This name refers to both **Pol**itical **Ca**reers and my key theoretical and empirical interest: **Pol**itical **Ca**pital.

This chapter exists out of two parts. In the first part, I summarise the rational of using a relational database and provide an general overview of the database as a whole. In the second part of this chapter I provide a more detailed overview of what information can be found in each of the different dataframes that together make up this database. After listing the variables, I also show snippets of the data. This helps to understand the setup of the database., also to allow readers of this chapter to start their own analysis, I include short examples of r-code that can be used (with suitable adjustments) to query the underlying data. This chapter is intended to be a first encounter with the PolCa data, and so focuses on providing a general overview. Those interested in the details should look at the codebook included in the appendix A to this thesis.

Part one: overall structure of the PolCa database

To ensure data quality and flexibility (see Chapter 2 of this thesis) the PolCa data is stored in a 'relational database' (Hernandez 1996). This means that one does not commit to a single data-format for statistical analysis. Instead, data is stored in an overarching flexible database from which analytical samples for statistical analysis can be generated. This approach is slightly more time-consuming, with regards to the setup. However, it comes with important benefits in terms of flexibility, reduced redundancy and enhanced reliability (see Chapter 2 for a detailed reflection on the conditions under which using a relational database is desirable). Answering the research questions asked in the empirical chapters that follow required four fundamentally different data-structures[1]. This made the decision to work from a central database crucial to completing the work needed for this thesis.

The core idea of a relational database is that information is not repeated across cells. Instead, information is stored once in a dataframe that contains everything there is to know about this entity. Together a set of connected dataframes forms one database. For example, we can see in Figure 1 that static information about the entity 'individuals' (POLI[2]) is stored separately from the (political) jobs or *resume entries* these individuals held throughout their careers (RESE) and that one individual can have multiple ('n') resume entries. It is these relations, that are depicted as lines in figure 1, that give relational databases their name. We build data ready for statistical analysis by combining and merging the information held in these dataframes together from across the database.

Figure 1 offers a representation of the PolCa database in a database Entity Relationship Diagram

[1] Chapter 4 uses 'opportunities for MPs to become cabinet members' as the key unit of analysis. Chapter 5 uses: the composition of 'factions / party groups' over time, *individual politicians at entry* and the aforementioned *cabinet opportunities*. Chapter 6 uses parliamentary episode data, where MPs occur as often in the data as they had seats in the Dutch parliament.

[2] Each dataframe in the PolCa database has a four-letter abbreviation. For reference purposes they have the colour of the blocks in Figure 1.

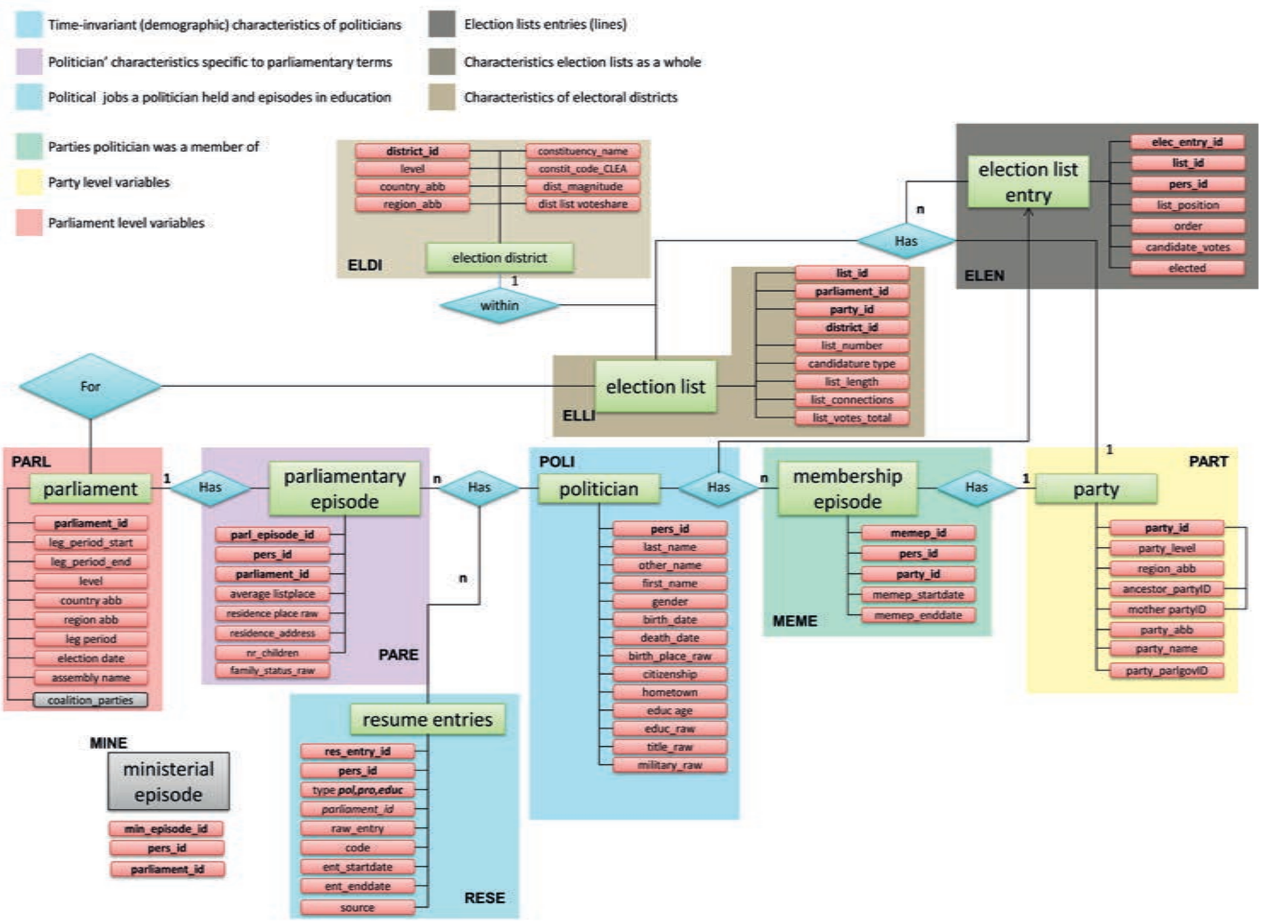


Figure 1: Entity Relationship Diagram of the PolCa data

(terminology from computer science, see Hernandez 1996). It provides a useful overview of the structure of the PolCa database. More specifically, it shows there are nine entities/dataframes in the database, each of with a focus on a different unit of analysis. The POLI, PARL, and PART dataframes contain the static characteristics of individual politicians, parliaments, and parties respectively. What politicians was a member of which parliament and party can be found in PARE and MEME. RESE contains all of the resume entries / jobs of politicians. Finally, ELLI, ELDI and ELEN contain information about election lists, election districts, and the entries of individual politicians on election lists. Figure 1 also shows the relationship between these entities.

As we can see from this overview, the presented database is set up with a specific focus on biographical data for individual politicians. Yet, whenever possible, links are provided to external data-sets that are maintained by other researchers. For example, the presented database can easily be combined with: the 'Dutch Parliamentary Voting Data-set' (Louwerse, Otjes and Vonno 2018), party-level data in 'parlgov' (Döring and Manow 2019) and the coded longitudinal ideological information from party manifestos as part of 'The Manifesto Project' (Volkens, Krause, Lehmann, Matthieß, Merz, Regel and Weßels 2019), among others.

The role of 'SQL Queries'

To extract data from the database in a desired end format that can be used as input for statistical analysis, a so-called 'SQL query' can be used. SQL (Structured Query Language) queries can be used to extract research question specific data from a relational database into data-frame suitable for statistical analysis. The recently released 'sqldf' package in R allows one to run SQL queries on R dataframes without the need to set up a database server.

For example, the piece of R-code[3] in code-block 1 will return output that tells us in which parliaments the current Dutch prime minister 'Mark Rutte[4]' and his right-wing populist colleague 'Geert Wilders[5]' have been active.

[3] The r-script used in this chapter, together with some sample data, can be found on https://github.com/-TomasZwinkels/R034.

[4] **Mark Rutte** was prime-minister (2010-2019) of the Netherlands. Earlier, he was junior minister of social affairs and subsequently for higher education. He has been the leader of his party, the VVD (People's Party for Freedom and Democracy) since 2006. He is known for his charismatic, though somewhat car salesman-like, demeanour.

[5] **Geert Wilders** is the leader of his own 'Party for Freedom'; he initially entered parliament as a member of the VVD but broke with that party in 2004. He worked as a speechwriter for the the VVD, earlier. His popularity is based on anti-immigration policies and anti-establishment rhetoric.

Code-block 1: Setup and example query

```
## setup

# packages required
    install.packages("sqldf")
        library(sqldf)

# data import
        POLI = read.csv("PCC/POLI.csv", header = TRUE, sep = ";")
        PARE = read.csv("PCC/PARE.csv", header = TRUE, sep = ";")

# politicians used in example
    peoplevec <- c("NL_Rutte_Mark_1967","NL_Wilders_Geert_1963")
    POLI <- POLI[which(POLI$pers_id %in% peoplevec),]

# query to get the parliaments these two MPS were in
sqldf("
    SELECT POLI.pers_id, POLI.birth_date, PARE.parliament_id
    FROM POLI LEFT JOIN PARE
    ON POLI.pers_id = PARE.pers_id
    ")
```

This code-block starts with some setup. The required r-packages are loaded, the relevant dataframes from the PolCa database are imported as .csv files and we focus on two cases. At the end of this code-block an SQL query is called. This is possible in R when the *sqldf* package is installed and loaded. This query combines (i.e. ‘left joins’, because information from the ‘right-side’ dataframe is added to the left-side dataframe) all politician level records (POLI) with available information on who was in which parliament (PARE). This query illustrates how SQL makes it easy to select and connect data from different dataframes. In this case three variables are selected from two different dataframes. The rows in these dataframes are matched by ‘pers_id’. Whenever information anywhere in the database refers to the same politician, a ‘unique identifier’ is used. This allows the SQL software to know what information should be combined. For reasons that I outline in Chapter 2, in the case of politicians, it worked best when a politician’s *first name*, *last name* and *birth year* were used. This code will then return the output below, shown in Data-output 1.

Data-output 1: output code-block 1: example query.

```
                 pers_id birth_date parliament_id
NL_Wilders_Geert_1963  06sep1963 NL_NT-TK_1998
NL_Wilders_Geert_1963  06sep1963 NL_NT-TK_2002
NL_Wilders_Geert_1963  06sep1963 NL_NT-TK_2003
NL_Wilders_Geert_1963  06sep1963 NL_NT-TK_2006
NL_Wilders_Geert_1963  06sep1963 NL_NT-TK_2010
NL_Wilders_Geert_1963  06sep1963 NL_NT-TK_2012
   NL_Rutte_Mark_1967  14feb1967 NL_NT-TK_2003
   NL_Rutte_Mark_1967  14feb1967 NL_NT-TK_2006
   NL_Rutte_Mark_1967  14feb1967 NL_NT-TK_2010
   NL_Rutte_Mark_1967  14feb1967 NL_NT-TK_2012
```

Part two: a detailed look at the data in each data-frame

I now continue with an overview of the data available in each dataframe within the PolCa database, and example sql queries that can be used to access this data. Table 1 provides an overview of these dataframes and the key sources that were used to construct them. It also mentions the extraction techniques - outlined in detail in Chapter 2 - that I used to construct these data.

Table 1: Overview of key sources and extraction techniques used per dataframe.

dataframe	**Key sources used**	**Key extraction techniques**
POLI: static individual	PDC* archive	Regular expressions**
PARE: episodes in parliaments	PDC archive	Regular expressions
PARL: parliaments	Staten Generaal Digitaal	Manual lookup
MEME: episodes in parties	Election list scans & PDC archive	Regular expressions & OCR***
PART: political parties	Election list scans & PDC archive	Regular expressions & OCR
ELEN: election list entries	Election list scans	Regular expressions & OCR
ELLI: election lists	Election list scans	Regular expressions & OCR
ELDI: election districts	Election list scans	Regular expressions & OCR
RESE: resume entries	PDC archive	Regular expressions, machine learning & CodeThing

*

Parliamentary Documentation Centre, see parlement.com.

** Regular expressions are advanced search language that can be used to extract patterned pieces of sub-text from a larger body of raw text. See Chapter 2 for details.

*** Optical Character Recognition: software that transforms scanned pictures into computer readable digital text. See Chapter 2 for details.

Static individual characteristics - POLI

The POLI[6] dataframe contains the static characteristics of individual politicians. These include personal identification labels and numbers as well as a politician's *first name*, *last name*, *gender*, *date of birth* and *birthplace*. Table 2 summarises the current state of these data.

Table 2: Summary of key elements of the politician (POLI) data, see codebook for details.

Variable	Description	N	% of all*	% elected**
pers_id	primary identifier	5983	100%	100%
id_nl_pdc[7]	identifier	3397	56.78%	100%
first_name	first name	5882	98.31%	100%
last_name	last name	5982	99.98%	100%
gender	gender	3543[8]	59.22%	100%
			(male: 83.40%)	(male: 85.55%)
birth_date	date of birth	3400	56.83%	99.97%
birth_place_raw	place of birth	3380	56.49%	99.5%

*percentage among all cases for which this information is available.
**percentage among MPs elected to parliament for whom this information is available.

The politicians in this sample are all national politicians. In the current state it contains all Dutch MPs and ministers between 1947 and 2012 and all candidates for the national parliament between 1982 and 2017. The included information comes from two merged sources: the digital archive of the Dutch Parliamentary Documentation centre (PDC, see www.parlement.com[9]) and election list data.

A look at POLI

Assuming we have just run the code in code-block 1, we can request the static individual information on POLI for Mark Rutte and Geert Wilders with the query in code-block 2.

Code-block 2: Setup

```
sqldf("
    SELECT POLI.pers_id, POLI.id_nl_pdc, POLI.first_name,
    POLI.last_name, POLI.gender, POLI.birth_date,
    POLI.birth_place_raw
    FROM POLI
    ")
```

[6] dataframes in this document are colour coded. The shown colour corresponds to the colour of the dataframe in Figure 1.

[7] This is the internal identification number that is used for this politician by the Dutch Parliamentary Documentation Center. Not all politicians in our data have been elected to parliament, hence not all politicians have a value for id_nl_pdc.

[8] Some of this information is only available for those who have been elected to parliament.

[9] State at June 2018.

This gives the following result:

Data-output 2 - from code-block 2: static politician level characteristics.

```
                pers_id id_nl_pdc first_name last_name gender birth_date birth_place_raw
1 NL_Wilders_Geert_1963      2258      Geert   Wilders      m  06sep1963           Venlo
2    NL_Rutte_Mark_1967      2396       Mark     Rutte      m  14feb1967    s-Gravenhage
```

We see that this data contains basic static information about the politicians. For example, their *name*, *gender* and *birth-date* (see the codebook for a complete list).

Who was in what parliament - PARE & PARL

Such static information is important, but to learn something about political career dynamics we need more. We might want to know *who was a member of which parliament*, for example, to measure how the percentage of women in parliaments has developed over time. This 'parliamentary episode' information is stored in PARE. The detailed resumes of MPs were used to build this dataframe[10]. An MP gets a PARE episode if she has been in a parliament for at least one day. The unit of analysis in this dataframe is thus episodes in parliament. Politicians occur as often in this dataframe as they have been re-(s)elected to the Dutch national parliament[11].

Table 3 summarises the state of the parliamentary episode dataframe (PARE).

Table 3: Summary of key elements of the PARE data, see codebook for details.

Variable	Description	N	% complete[12]
parl_episode_id	primary identifier	3497	100%
pers_id	person identifier, POLI	3497	100%
parliament_id	parliament identifier, PARL	3497	100%

Now that we know who was in which parliament we can also add information on the level of the parliament. These data-points are stored in PARL. This information comes from the Parliamentary Documentation center, the 'Jaarboek parliamentaire geschiedenis' (Yearbook Parliamentary History), Wikipedia and 'Staten Generaal Digitaal' (Dutch 'Hansard' which contains a verbatim report of the proceedings of both Dutch houses). Users can request a parliament's *start-* and *end date*, *date of election*, whether this election was a *regular or snap election* and what political parties were in this parliament's *governing coalition*. Table 4 summarises the current state of this data.

[10] Special thanks to research assistant Adrian Sutter for his relentless efforts in that direction.

[11] The data-structure is set up such that episodes in other elected positions, like regional or municipal parliaments, can also be added to this dataframe. In the current version of the data this has not yet been done.

[12] 'Complete' refers to the percentage of all records for which a value on this variable is available.

Table 4: Summary of key elements of the parliamentary (PARL) data, see codebook for details.

Variable	Description	N	% complete
parliament_id	primary identifier	15	100%
election_date	date of election	15	100%
election_type	snap election or not	15	100%
coalition_parties	list of parties in government coalition	15	100%

A look at parliaments and who was in them: PARE & PARL

When this information is combined, we know who was in which parliament. We also obtain some important additional contextual information. Assuming that the code in the code-blocks so far has been run, code-block 3 shows how this data can be requested.

Code-block 3: PARE and PARL example

```
POLI <- sqldf("
        SELECT POLI.pers_id, PARL.parliament_id, PARL.election_date,
        PARL.election_type, PARL.coalition_parties
        FROM POLI LEFT JOIN PARL
        ON
        POLI.parliament_id = PARL.parliament_id
        ")
```

The resulting output looks like this:

Data-output 3, from code-block 3, parliamentary membership and parliament level characteristics.

```
                 pers_id parliament_id election_date election_type              coalition_parties
1  NL_Wilders_Geert_1963 NL_NT-TK_1998     06may1998       regular NL_PvdA_NT;NL_VVD_NT;NL_D66_NT
2  NL_Wilders_Geert_1963 NL_NT-TK_2002     15may2002         early  NL_CDA_NT;NL_LPF_NT;NL_VVD_NT
3  NL_Wilders_Geert_1963 NL_NT-TK_2003     22jan2003         early  NL_CDA_NT;NL_VVD_NT;NL_D66_NT
4  NL_Wilders_Geert_1963 NL_NT-TK_2006     22nov2006         early  NL_CDA_NT;NL_PvdA_NT;NL_CU_NT
5  NL_Wilders_Geert_1963 NL_NT-TK_2010     09jun2010         early            NL_VVD_NT;NL_CDA_NT
6  NL_Wilders_Geert_1963 NL_NT-TK_2012     12sep2012         early           NL_VVD_NT;NL_PvdA_NT
7     NL_Rutte_Mark_1967 NL_NT-TK_2003     22jan2003         early  NL_CDA_NT;NL_VVD_NT;NL_D66_NT
8     NL_Rutte_Mark_1967 NL_NT-TK_2006     22nov2006         early  NL_CDA_NT;NL_PvdA_NT;NL_CU_NT
9     NL_Rutte_Mark_1967 NL_NT-TK_2010     09jun2010         early            NL_VVD_NT;NL_CDA_NT
10    NL_Rutte_Mark_1967 NL_NT-TK_2012     12sep2012         early           NL_VVD_NT;NL_PvdA_NT
```

This data for example reveals that ‘Mark Rutte’ was never elected to parliament after a regular election but rather in early ‘snap’ elections.

Who was a member of which political party when - MEME & PART

Politicians typically represent a specific political party at a specific point in time. Sometimes they switch alliances. *Who was a member of which political party when* can be found - in combination with some important information on these parties - in the dataframes MEME (MEMbership Episodes) and PART. This information is based on the Parliamentary Documentation Centre archive. This information was crossed-checked with election list data[13]. Tables 5 and 6 summarise the current state of these data.

Table 5: Summary of key elements of the party (PART) data, see codebook for details.

Variable	Description	N	% complete
party_id	primary identifier	63	100%
ancestor_party_id	list of parties that party came out of	63	100%
party_name	name(s) of party	62	98.41%
party_parlgov_id	external identifier	63	100%
episode_start	when was party founded (if so)	62	98.41%
episode_end	when was party formally dissolved (if so)	61	96.83%

Table 6: Summary of key elements of the party membership episodes (MEME) data, see codebook for details.

Variable	Description	N	% complete
memep_id	primary identifier	1592	100%
pers_id	person identifier, POLI	1592	100%
party_id	party identifier, PART	1592	100%
memep_startdate	when did membership start	1592	100%
memep_enddate	when did membership end (if so)	1592	100%

The PART dataframe contains important information about the political party, like the *successor* or *ancestor* parties, identifiers that link this party to existing external databases with party level information and information on when the party was *founded* and (potentially) *dissolved*. MEME contains the information on *which politician was a member of which political party when*. To make sure that we can derive this information, politicians who switch parties once or more or who have held the membership of multiple political parties (for example a small local party as well as a national party) will occur multiple times in this dataframe.

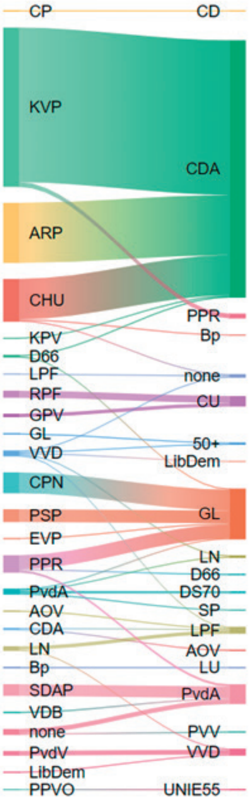
A look at party membership MEME & PART

Figure 2 shows a so-called 'Sankey diagram' of these 'switches'[14]. We see the old party, of which the politician was a member, on the left side of the figure, and the new party, which that politician became a member of, on the right side.

[13] With special thanks to Oliver Huwyler for providing the procedures and r-scripts to do so.

[14] When parties merge, like the Dutch Christian Demographic party (CDA) in 1983, then the members 'switch' with them.

Figure 2: Collective and individual party switching by Dutch MPs (1947-2012)[15]



[15] 50+: 50PLUS, AOV: General Senior Union, ARP: Anti Revolutionary Party, BP: Farmers Party, BVL: League of Free Liberals, CD: Centre Democrats, CDA: Christian Democratic Appeal, CDU: Christian Democratic Union, CHU: Christian Historical Union, CP: Centre Party, CP86: Centre Party '86, CPN: Communist Party of the Netherlands, CU: ChristianUnion, D66: Democrats 66, DS70: Democratic Socialists 70, EB: Economic League, GL: GreenLeft, GPV: Reformed Political League, Groen: The Greens, GW: Wilders Group, HGS: New Reformed State Party, KNP: Catholic National Party, KVP: Catholic Peoples Party, LidDem: Liberal Democratic Party, LN: Livable Netherlands, LPF: Fortuyn List, LSP: The Freedom League, LU: Liberal Union, PB: Peasants' League, PPR: Radical Political Party, PSP: Pacifist Socialist Party, PvdA: Labour Party, PvdV: Freedom Party, PVV: Party for Freedom, RKP: Roman Catholic Party, RKSP: Roman-Catholic Political Party, RKVP: Roman Catholic People's Party, RPF: Reformatory Political Federation, RSP: Revolutionary Socialist Party, SDAP: Social Democratic Workers' Party, SDP: Social-Democratic Party, SP: Socialist Party, UNIE55: Unie 55+, VDB: Free-thinking Democratic League, VSP: United Seniors Party, VVD: People's Party for Freedom and Democracy.

The two queries below (code-block 4 and 5) continue our analysis to request this information for our two focus cases. What makes this code-block particularly interesting is that it shows how date-ranges can be used to merge the correct information together. The time-sensitive nature of political career data makes it crucial to be able to use date information to extract and recombine data.

Code-block 4: MEME example

```
TEMP <- sqldf("
        SELECT POLI.pers_id, POLI.parliament_id, POLI.election_date,
        MEME.party_id, MEME.memep_startdate
        FROM POLI LEFT JOIN MEME
        ON
               POLI.pers_id = MEME.pers_id
               AND
                       (
                       POLI.election_date >= MEME.memep_startdate
                       AND
                       POLI.election_date <= MEME.memep_enddate
                       )
       ")
```

The resulting output is as follows:

Data-output 4 - from code-block 4: party membership over time.

```
                 pers_id parliament_id election_date  party_id memep_startdate
1  NL_Wilders_Geert_1963 NL_NT-TK_1998     06may1998 NL_VVD_NT       01jan1989
2  NL_Wilders_Geert_1963 NL_NT-TK_2002     15may2002 NL_VVD_NT       01jan1989
3  NL_Wilders_Geert_1963 NL_NT-TK_2003     22jan2003 NL_VVD_NT       01jan1989
4  NL_Wilders_Geert_1963 NL_NT-TK_2006     22nov2006 NL_PVV_NT       22feb2006
5  NL_Wilders_Geert_1963 NL_NT-TK_2010     09jun2010 NL_PVV_NT       22feb2006
6  NL_Wilders_Geert_1963 NL_NT-TK_2012     12sep2012 NL_PVV_NT       22feb2006
7     NL_Rutte_Mark_1967 NL_NT-TK_2003     22jan2003 NL_VVD_NT       01jan1988[[lcen]]
8     NL_Rutte_Mark_1967 NL_NT-TK_2006     22nov2006 NL_VVD_NT       01jan1988[[lcen]]
9     NL_Rutte_Mark_1967 NL_NT-TK_2010     09jun2010 NL_VVD_NT       01jan1988[[lcen]]
10    NL_Rutte_Mark_1967 NL_NT-TK_2012     12sep2012 NL_VVD_NT       01jan1988[[lcen]]
```

We learn that Mark Rutte consistently was a member of the VVD (conservaties), while Geert Wilders left the VVD for another new party[16]. We also see that it is not known exactly when Rutte became a member of the VVD. This start-date is as such 'left censored'[17].

A look at the party data in PART

We now know who was a member of which party. This also enables us to subsequently merge in party level characteristics. Code-block 5 does this.

[16] In fact, he founded this right-wing populist party.

[17] indicating that we know that he was probably a member earlier then the mentioned date, although we do not know for how long.

Code-block 5: MEME example

```
sqldf("
    SELECT TEMP.pers_id, TEMP.parliament_id, PART.party_parlgov_id
    FROM TEMP LEFT JOIN PART
    ON TEMP.party_id = PART.party_id
        ")
```

This adds additional information to this dataframe at the party level.

Data-output 5 - from code-block 5: party membership and party characteristics.

```
               pers_id parliament_id  party_id                           party_name  parlgov_id
1  NL_Wilders_Geert_1963 NL_NT-TK_1998 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
2  NL_Wilders_Geert_1963 NL_NT-TK_2002 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
3  NL_Wilders_Geert_1963 NL_NT-TK_2003 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
4  NL_Wilders_Geert_1963 NL_NT-TK_2006 NL_PVV_NT                 Partij voor de Vrijheid        1501
5  NL_Wilders_Geert_1963 NL_NT-TK_2010 NL_PVV_NT                 Partij voor de Vrijheid        1501
6  NL_Wilders_Geert_1963 NL_NT-TK_2012 NL_PVV_NT                 Partij voor de Vrijheid        1501
7     NL_Rutte_Mark_1967 NL_NT-TK_2003 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
8     NL_Rutte_Mark_1967 NL_NT-TK_2006 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
9     NL_Rutte_Mark_1967 NL_NT-TK_2010 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
10    NL_Rutte_Mark_1967 NL_NT-TK_2012 NL_VVD_NT Volkspartij voor Vrijheid en Democratie        1409
```

One key bit of information that can be added like this is the party's so-called '*parlgov_id*'. This numerical identifier allows one to merge in a lot of additional information, for instance a party's left-right position, from the external parlgov database [18] and connected data-sets for example the manifesto data[19] with longitudinal policy positions and institutional data from the Integrated Party Organisation Dataset[20].

Who (ELEN) was on what election list (ELLI) when and where (ELDI)

Next to knowing who was associated with what party, electoral information, such as *who was running on what list-position in which electoral district(s)*, is equally crucial. Election list data contain a whole variety of information. We can use this to identify who was nominated by what party in what districts. It also tell us what information voters saw when choosing between politicians, for example in what order candidates occurred. Finally, the included election list data also contains disaggregated electoral outcomes, so we can find out precisely how many people voted for which politician in what district.

Following the relational database philosophy of storing information at the natural level / unit of analysis it occurs in, I store this information in three dataframes. ELEN contains election list entries and their characteristics, such as which politician held this position on the list and how many votes they got. ELLI contains all information on the level of the election list, for example which party submitted this list. ELDI contains the election districts, including the name of the district in

[18] See parglov.org

[19] https://manifesto-project.wzb.eu/

[20] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/PE8TWP

the Constituency-Level Elections Archive (CLEA). This allows key electoral information from their archive to be merged in; for example the number of eligible voters in a district.

I extracted this information from scans of official election outcomes ('process verbaal'(official legal report)) that were generously made available to me by the Dutch National Election Counsel ('Kiesraad')[21] and the Dutch national archive. These .pdf files were digitised and tabulated using Optical Character Recognition (OCR) with the software Abby FineReader (version 12.0)[22]. All extractions were subsequently combined, checked[23] and merged into the central database using the procedures outlined in Chapter 2. This information is currently available from 1982 onward. Information for the following smaller political parties is not yet included in this data due to time-constraints: 50+, AOV, CD, CPN, CP, CU, EVP, GPV, LN, LPF, PPR, PSP, PvdD, RPF, SGP UNIE55. Together, these currently excluded parties held 132 of the seats (12.1%) in the Dutch parliament between 1982 and 2012.

The tables 7, 8 and 9 below summarise the state of the currently available data.

Table 7: Summary of the election list entries (ELEN) data (1982 - 2012)

Variable	Description	N	% complete
pers_id	person identifier, POLI	56581	~87.9%[24]
district_id	district identifier, ELDI	56581	~87.9%
listplace	position on election list	56581	~87.9%
candidate_votes_district	number of votes in each district	19246	34.0%
candidate_votes_national	total number of votes	54762	96.8%

Table 8: Summary of key elements of the election list (ELLI) data (1982-2012), see codebook for details.

Variable	Description	N	% complete
list_id	primary identifier	1264	~87.9%
list_name	name of list / party	1264	~87.9%
parliament_id	parliament identifier, PARL	1264	~87.9%
district_id	district identifier, ELDI	1264	~87.9%
party_id	party identifier, PART	1264	~87.9%
list_length	number of people on list	1248	~87.9%

[21] Special thanks to Ron de Jong.

[22] Special thanks for this work goes to - then student assistant - Renske Verweij.

[23] Special thanks for this goes to Niels Goet.

[24] This number is an estimate. For all the election list entries currently in the data this information is available; we have, however, also missing data from 12.1% of all seats over the observation period. Because not all parties submit lists of the same length, nor do they submit lists in all districts, it is not possible to give an exact estimate at this stage.

Table 9: Summary of key aspects of the election district (ELDI) data (1947-2012), see codebook for details.

Variable	Description	N	% complete
district_id	primary identifier	506	100%
region_abb	two-letter abbreviation of district	506	100%
district_aliases	label commonly used for district (1-20)	413	81.62%
constituency_name	name of district	506	100%
constituency_name_CLEA	name of district in the CLEA data	506	100%

A look at the election lists ELEN, ELLI & ELDI

Election data encompasses three interrelated units of analysis. These are 'election list', 'election list entries' and 'election districts'. Following the philosophy of a relational database, I store this data in three different dataframes. Election list data is stored in ELLI, an election list is a table of names, in a certain order. ELLI contains information that pertains to election lists as a whole. Two key examples of this are the list *electoral district* ('kies-kring') and the political *party* that submitted this list. The second dataframe contains election list entries. They are stored in ELEN. Each row in ELEN describes one row of an election list. Key information here is for example a politician's list position or how many votes a politician got. The code in code-block 6 combines the information from these two dataframes and outputs some example data.

Code-block 6: ELEN example

```
# get ELEN entries for Rutte and Wilders only
        ELENEX  <- ELEN[which(ELEN$pers_id %in% peoplevec ),]

# first we merge ELEN and ELLI together
        EL <- sqldf("
                     SELECT ELENEX.*, ELLI.*
                                 FROM ELENEX LEFT JOIN ELLI ON
                             ELENEX.list_id = ELLI.list_id
                                ")

# just some data preview, every 10th row.
        sqldf("SELECT pers_id,district_id,listplace
                 FROM EL ORDER BY pers_id")[c(seq(from=0,to=200,by=10)),]
```

This generates the following output:

Data-output 6 - from code-block 6: some election list entries.

```
                  pers_id                       district_id listplace
10      NL_Rutte_Mark_1967            NL_NT-TK_2003__Haarlem        11
20      NL_Rutte_Mark_1967   NL_NT-TK_2003__Netherlands[1-5]        11
30      NL_Rutte_Mark_1967             NL_NT-TK_2010__Arnhem         1
40      NL_Rutte_Mark_1967            NL_NT-TK_2010__Tilburg         1
50      NL_Rutte_Mark_1967             NL_NT-TK_2012__Arnhem         1
60      NL_Rutte_Mark_1967            NL_NT-TK_2012__Tilburg         1
70      NL_Rutte_Mark_1967            NL_NT-TK_2006__Haarlem         1
80      NL_Rutte_Mark_1967     NL_NT-TK_2006__sHertogenbosch         1
90  NL_Wilders_Geert_1963             NL_NT-TK_1998__Arnhem        45
100 NL_Wilders_Geert_1963            NL_NT-TK_1998__Tilburg        45
110 NL_Wilders_Geert_1963             NL_NT-TK_2002__Zwolle        30
120 NL_Wilders_Geert_1963          NL_NT-TK_2002__Dordrecht        30
130 NL_Wilders_Geert_1963          NL_NT-TK_2003__Groningen        14
140 NL_Wilders_Geert_1963          NL_NT-TK_2003__DenHelder        14
150 NL_Wilders_Geert_1963 NL_NT-TK_2003__Netherlands[12-16]        14
160 NL_Wilders_Geert_1963             NL_NT-TK_2006__Arnhem         1
170 NL_Wilders_Geert_1963            NL_NT-TK_2006__Tilburg         1
180 NL_Wilders_Geert_1963             NL_NT-TK_2010__Arnhem         1
190 NL_Wilders_Geert_1963            NL_NT-TK_2010__Tilburg         1
200 NL_Wilders_Geert_1963             NL_NT-TK_2012__Arnhem         1
```

The third dataframe contains information on the electoral district, this data is stored in ELDI. The Netherlands currently has 20 electoral districts. Before 2002 the country had 19 districts. Such information is stored in ELDI. Similar to the PART data, ELDI also contains an important 'external' identifier, in this case the *constituency code* in the 'CLEA' election data archive[25]. The external CLEA data contains a variety of potentially relevant information, for example the number of eligible voters in a district and the voter turnout.

The code in code-block 7 sums up these data for each election for our two example cases.

[25] http://www.electiondataarchive.org/

Code-block 7: Aggregated data from ELEN, ELLI and ELDI example

```
# then we get the info from ELDI in as well
        ELL <- sqldf("
                      SELECT EL.*, ELDI.*
                          FROM EL LEFT JOIN ELDI ON
                                  EL.district_id = ELDI.district_id
                                 ")

# make a new variable that contains person and parliament for grouping results
        ELL$parl_episode_id_f <- paste(ELL$pers_id,ELL$parliament_id,sep="__")

# output the result
        sqldf("
                SELECT ELL.pers_id, ELL.parliament_id, ELL.party_id,
                    SUM(ELL.candidate_votes) as `total_votes',
                    AVG(ELL.listplace) as `average_list_position'
                    FROM ELL
                    GROUP BY parl_episode_id_f
                  ")
```

This generates the following output:

Data-output 7 - from code-block 7: aggregated election list data.

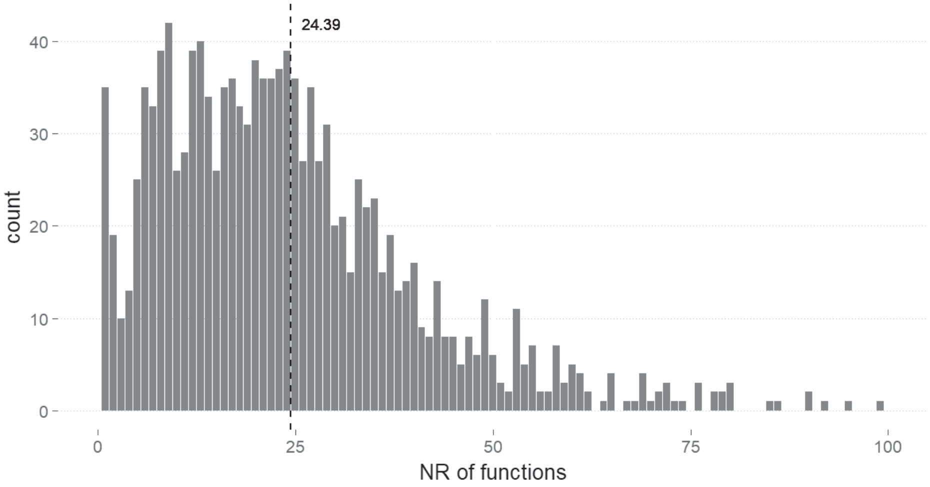
```
                 pers_id parliament_id  party_id total_votes average_list_position
1     NL_Rutte_Mark_1967 NL_NT-TK_2003 NL_VVD_NT        4297                    11
2     NL_Rutte_Mark_1967 NL_NT-TK_2006 NL_VVD_NT      553200                     1
3     NL_Rutte_Mark_1967 NL_NT-TK_2010 NL_VVD_NT     1617636                     1
4     NL_Rutte_Mark_1967 NL_NT-TK_2012 NL_VVD_NT     2129000                     1
5  NL_Wilders_Geert_1963 NL_NT-TK_1998 NL_VVD_NT         334                    45
6  NL_Wilders_Geert_1963 NL_NT-TK_2002 NL_VVD_NT        2522                    30
7  NL_Wilders_Geert_1963 NL_NT-TK_2003 NL_VVD_NT        4763                    14
8  NL_Wilders_Geert_1963 NL_NT-TK_2006 NL_PVV_NT      566197                     1
9  NL_Wilders_Geert_1963 NL_NT-TK_2010 NL_PVV_NT     1376938                     1
10 NL_Wilders_Geert_1963 NL_NT-TK_2012 NL_PVV_NT      886314                     1
```

By having access to this information we can now see that Geert Wilders only received 334 votes in 1998 (the year when he was first elected to parliament). We can also see that both politicians only started to receive a substantial number of votes after they became party leader ('lijsttrekker', list position: 1). Such information can be of great value in a variety of analyses, as Chapter 6 of this thesis in particular illustrates.

Who had what (political) job when - RESE

Finally, for political career research to live up to its full potential (see Chapter 1), we need to know *who had what (political) job when*, consequently, a core part of the PolCa data concerns the resume entry (RESE) dataframe. This dataframe contains a long list of all functions (political and non-political jobs

Figure 3: Number of functions in the data for Dutch MPs (1947-2012)



and side-functions) that Dutch MPs held throughout their (political) career. Functions are defined as *all positions politicians can hold*. This entails paid and unpaid functions as well as full-time and part-time ones. It can be as prestigious as being a prime minister and as small as a trivial voluntary activity for a local sports club. Everything that entails an activity that politicians consider worth reporting on their resume is included. Each politician occurs multiple times in this data. A politician who held a total of 20 different (political) functions throughout that individual's career will occupy 20 lines in this dataframe. Figure 3 shows the frequency distribution of the total number of functions. We can see that Dutch MPs on average hold 24 functions during their entire[26] political career.

This data is based on the online biographical archive of the Parliamentary Documentation Centre[27], to which I was kindly given access. I assigned career labels / coded all of the data from this archive. As such, the data can be seen as an 'enriched' machine-readable version of the parliamentary documentation centre archive.

The online software 'CodeThing' (CodeThing.org) was used to manually code all the functions. This software was especially developed for this purpose[28]. This was a major task that could only be achieved through months of work by successive coding teams [29]. The important result of all of this work is that the resume entry data in RESE, which was derived from the PDC archive, can now be

[26] As some of the politicians in our data have not concluded their careers yet, the actual total number is probably even a bit higher.

[27] A special thanks goes to Bert van de Braak - Chief Editor of the archive - for a thorough introduction to the archive and many kind and quick responses to my questions.

[28] Many thanks to my brother Tijs Zwinkels (TinkerTank), who developed this software.

[29] A big thanks is in order to many people. In order of time and energy invested for the Dutch data they are: Joyce van de Schootbrugge, Renske Verweij, Oliver Huwylyer, Adrian Sutter, Elena Frech and Tobias Gysin.

used for quantitative statistical analysis.

On the most general level, this dataframe distinguishes between two types of jobs. *Political functions* are functions that are directly or indirectly aimed at creating and shaping policy. The remaining professional *non-political functions* are functions that do not have such a goal. An example of a political function is being an elected representative or working for a labour union. An example of a non-political function is working as an independent lawyer or as a secondary school teacher.

All non-political functions are coded into the **existing** ISCO-08 coding scheme. All political functions are coded into a **new** coding scheme that I developed together with colleagues (among others Melinda Mills, Renske Verweij, Joyce van de Schootbrugge, Oliver Huwyler[30], Elena Frech, Stefanie Bailer, Philip Manow, Simon Hug and Wang Leung Ting; see the codebook in the appendix for detailed specifications).

Table 10 and 11 below summarise the state of the RESE data for the *professional* (non-political) and *political* functions, respectively. For the non-political jobs Figure 4 shows the distribution of the main 'first digit' category from the international standardised code of occupations (ISCO08, see ILO, 2012)). The figures 5,6,7,8 and 9 show the distribution of jobs across some of the most important elements of a new, detailed political jobs coding scheme.

Non-political jobs

Table 10: Summary of key elements of the professional jobs in the resume entry data (RESE) (1947-2012), for details see codebook.

Variable	Description	N	% complete
res_entry_id	primary identifier	3721	100%
pers_id	person identifier, POLI	3721	100%
res_entry_start	date when function started	3720	99.97%
res_entry_end	date when function ended	3720	99.97%
res_entry_raw	text description as provided by source	3721	100%
policy_area	related policy area (CAP[31])	3537	95.06%
isco08	occupational code	3719	99.95%

We can see in Figure 4 that of the 3721 non-political jobs which the politicians in my data held, by far the majority were in the relatively high status category of ('technicians and associate professionals'). Examples of common jobs in this category are: lawyer, policy professional and journalist.

[30] Oliver Huwyler in particular deserves credit for his unremitting efforts towards helping me to narrow down and optimise my poorly thought through earlier coding schemes.

[31] Comparative Agendas Project, see https://www.comparativeagendas.net/

Figure 4: Distribution of professional functions (n=3719) in the RESE data, ISCO-08 main categories (1947-2012)

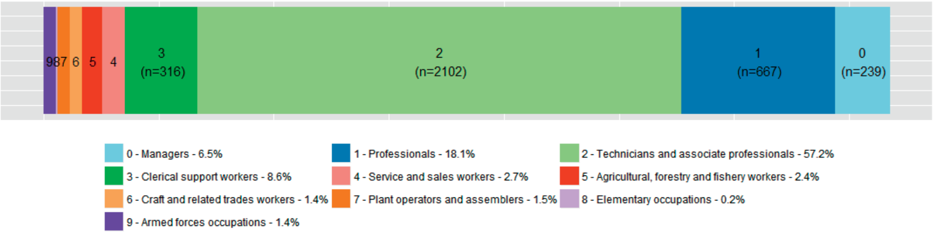


Table 11: Summary of key elements of the political jobs in (RESE) (1947-2012), see codebook for more details.

Variable	Description	N	% complete
res_entry_id	primary identifier	27136	100%
pers_id	person identifier	27136	100%
res_entry_start	date when function started	17846	65.77%
res_entry_end	date when function ended	15383	56.69%
res_entry_raw	text description as provided by source	27136	100%
pf_geolevel	geographical level	20237	74.58%
pf_instdomain	institutional domain	26368	97.17%
pf_orglevel	tier in the organisational hierarchy	24068	88.69%
pf_policy_area	related policy area (CAP)	20492	75.52%
pf_position	type of position	25077	92.41%

Political jobs

Next to ‘regular’ non-political jobs (such as working as a teacher), MPs can also hold political jobs, for example a board function for an interest group or political party or an elected political position. Political functions in the data are coded on the basis of five categorical variables[32]. First, jobs can occur at different *levels* in the political system (e.g. municipal or national level). Second, they can be in a different *environment* (e.g. legislative, for a party or an interest group). Third, a job can occur at a different *tier* in an organisation’s hierarchy (e.g. board of a party). Fourth, jobs can be associated with a specific *policy area*, like transport or education, or not (being a member of a municipal council by itself for example is not associated with expertise in a specific policy area). Fifth and finally, one can hold a specific kind of *position*, like a committee-chair, vice-chair or a regular member. Almost[33]

[32] Many thanks to Oliver Huwyler for co-developing this coding scheme.

[33] Coding work with a data-set of this size is never completely done. We can see in Table 11 that despite my best efforts, even at this stage some coding work still remains; in many cases this involves incomplete job specifications that require extensive archive research and/or personal contact with (ex)MPs. This remains to be done.

all political functions in the data have been coded into all of these five categories when applicable. It is noteworthy that specific combinations of values on these variables often uniquely signify concrete positions; the combined *political function code* 'NT_LE-LH_T3_NA_03' for example marks all political functions that entail membership of the Dutch national parliament. We know this because only this unique position meets the characteristics of being a national (NT) legislative (LE) position in the Dutch lower house (-LH) on the lowest tier (T3) as a regular member (03).

Level

Figure 5 shows the level at which political positions in the data occur. It is apparent that the great majority of these positions occur at the national level.

Figure 5: Distribution of *level* at which political jobs occur in the RESE dataframe, N=20,237 (1947-2012)

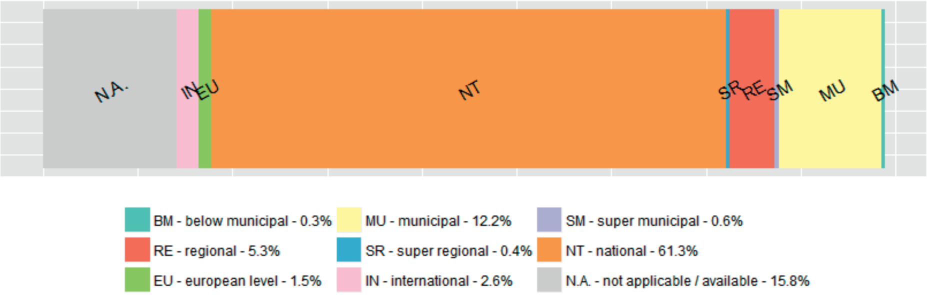
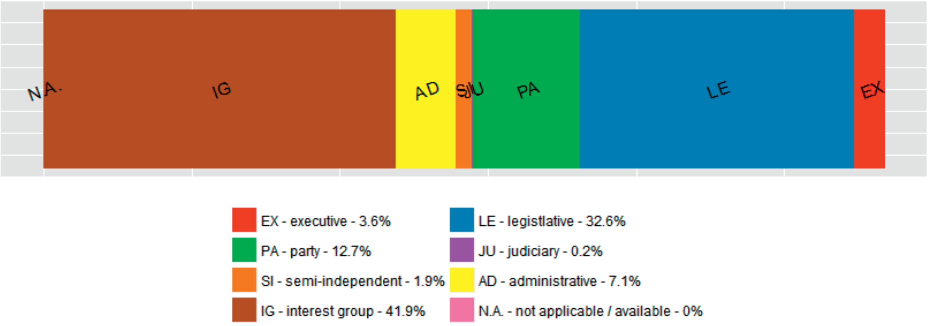


Figure 6 displays the breakdown of political jobs across the type of environment. We can see that the majority of jobs are in either the legislature (for example in a local, regional or national parliament), for an interest group (like a labour union, a confederation of industries or employers, e.g. 'VNO-NCW') or for a political party (for example on its board or in one of its committees).

Environment

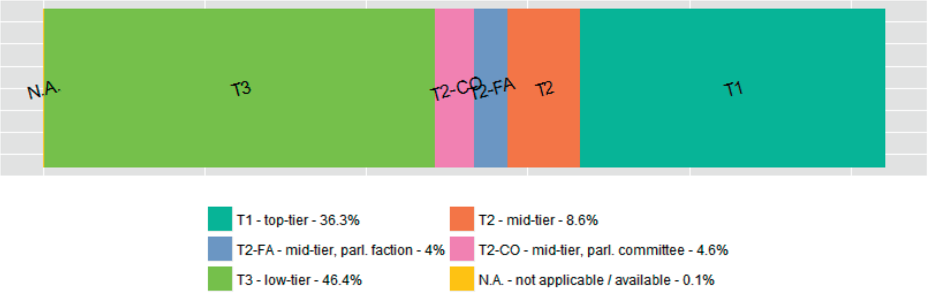
Figure 6: Distribution of *environment* of political position in the RESE dataframe, N=28,387 (1947-2012)



Tier

Figure 7 shows the breakdown by position in the organisational hierarchy. We can see that parliamentarians often occupy either the top or the bottom tier of the organisations they are active for. This large share of 'tier one' positions is interesting. It suggests that if MPs are active for an organisation, they typically are part of this organisation's decision-making elite. This can be seen as a reflection of the relatively high status this selected group of individuals are accustomed to throughout their (political) careers.

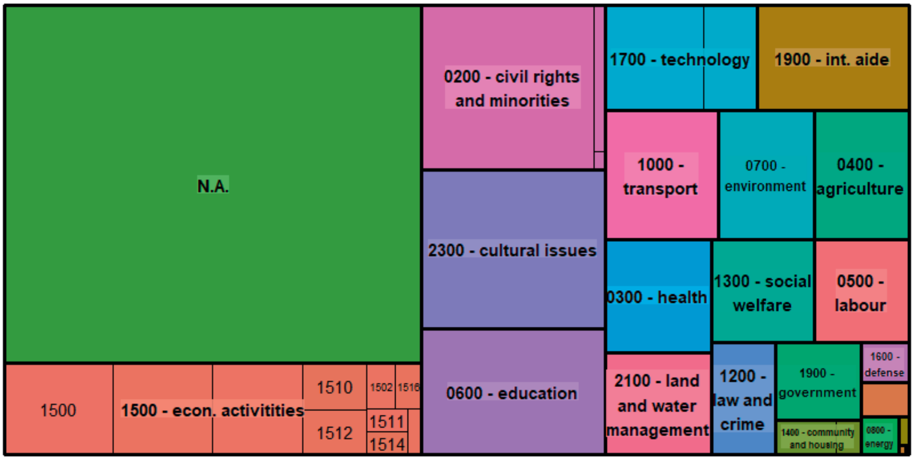
Figure 7: Distribution of the tier of political positions in the tier of the organisational hierarchy in the RESE dataframe, N=23,368 (1947-2012)



Policy area

Politicians are typically expected to hold a certain policy expertise. Figure 8 shows the specific policy areas that the functions that MPs held are related to. For example, a function for a climate change related interest group was coded as '1900 - environment', and if someone was a minister of education this would be marked as '0600 - education'. Functions can be related to multiple policy areas. Being the dean of an academic hospital for example would be marked as entailing expertise in both '0300 - health' and '0600 - education'. The so-called 'tree diagram' in Figure 8 shows that for a large proportion of functions no associated policy area can be defined (N.A., for Not Applicable). An episode in a local parliament for example is coded as - by itself[34] - not coming with any particular policy expertise. Of the political functions that do, the majority come with business expertise (1500) or are associated to civil-rights (0200), cultural issues (2300) or education (0600).

Figure 8: Distribution of comparative agenda *policy areas* in the RESE dataframe, when applicable N=32,867 (1947-2012)

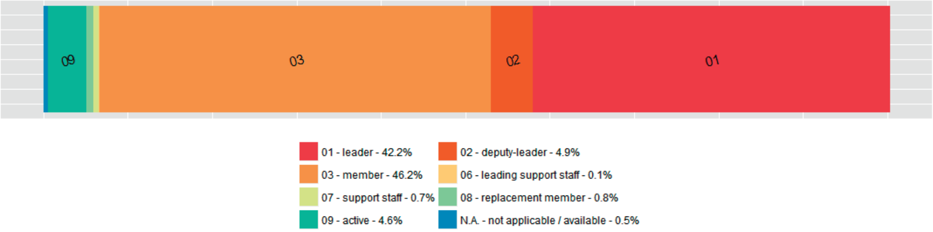


Position

The last coded political job variable captures the position of a politician within the job, for example as either a regular member (03), chair (01) or a mere support staff (07). We can see in Figure 9 that the MPs in this data very often are the chair / leaders of the organisations they are active for.

[34] If a politician was active in a substantive committee at the same time, for example committee on spatial planning, then that committee would be added to the data in a separate row and a related policy area would be specified.

Figure 9: Distribution of *positions* RESE dataframe, N=27,086 (1947-2012)



A look at the careers of Mark Rutte and Geert Wilders

On the basis of this coded career data a relatively detailed image can be constructed of the careers of (groups of) specific politicians. Code-block 8 below shows how to request this information for our two example politicians.

Code-block 8: RESE example

```
# select variables to display and order the result by pers_id and then date

        RESE <- sqldf("
                        SELECT pers_id,type,start,end,res_entry_raw,level,
                        environment,tier,policy_area,position
                        FROM RESE
                        ORDER BY pers_id, res_entry_start_p DESC
                        ")

        # focus on our two culprits
                RESEEX  <- RESEEX[which(RESEEX$pers_id %in% peoplevec ),]

        # display the results for each politician individually
                RESEEX[which(RESEEX$pers_id == "NL_Rutte_Mark_1967"),]
                RESEEX[which(RESEEX$pers_id == "NL_Wilders_Geert_1963"),]
```

The resulting output is displayed on the next page. We can see that the career data in the PolCa database is quite extensive and coded to a high level of detail. When we look at Mark Rutte's career for example it becomes apparent that, before his first legislative function ('NT_LE') as a member of the national parliament, he was a junior minister ('NT_EX'_T3). We can also see that before that he worked in the private sector ('1515'). Geert Wilders's first legislative function ('LE') was at the local level (MU). At that time he was also support staff ('07') at a faction (T3-FA) in the lower house ('-LH') of the national ('NT') parliament ('LE').

Data-output 8 - from code-block 8: career data.

```
           pers_id type     start       end                                                                                         res_entry_raw level environment  tier policy_area position
NL_Rutte_Mark_1967  pol 23mar2017 26oct2017                            lid Tweede Kamer der Staten-Generaal van 23 maart 2017 tot 26 oktober 2017    NT       LE-LH    T3        <NA>       01
NL_Rutte_Mark_1967  pol 20sep2012   oct2012                                          lid Tweede Kamer der Staten-Generaal vanaf 20 september 2012    NT       LE-LH    T3        <NA>       01
NL_Rutte_Mark_1967  pol 20sep2012 05nov2012                        lid Tweede Kamer der Staten-Generaal van 20 september 2012 tot 5 november 2012    NT       LE-LH    T3        <NA>       01
NL_Rutte_Mark_1967  pol 13sep2012   oct2012                        fractievoorzitter VVD Tweede Kamer der Staten-Generaal vanaf 13 september 2012    NT       LE-LH    T1        <NA>       01
NL_Rutte_Mark_1967 elec 04may2012 12sep2012                   lijsttrekker VVD Tweede Kamerverkiezingen 2012 van 4 mei 2012 tot 12 september 2012    NT       PA-MA    T1        <NA>       01
NL_Rutte_Mark_1967  pol 14oct2010   aug2012                               minister-president en minister van Algemene Zaken vanaf 14 oktober 2010    NT          EX    T1          NC       01
NL_Rutte_Mark_1967  pol 07oct2010 14oct2010  kabinetsformateur van 7 oktober 2010 tot 14 oktober 2010 (kreeg zijn opdracht kort voor het begin va    NT          LE    NC          NC       NC
NL_Rutte_Mark_1967 elec 12mar2010 09jun2010                      lijsttrekker VVD Tweede Kamerverkiezingen 2010 van 12 maart 2010 tot 9 juni 2010    NT       PA-MA    T1        <NA>       01
NL_Rutte_Mark_1967  pol 15sep2006   oct2010  gastdocent Intercollege Business School te `s-Gravenhage van 15 september 2006 tot oktober 2010 (1 d  <NA>          IG    T3        0600       09
NL_Rutte_Mark_1967  pol 29jun2006 08oct2010            fractievoorzitter VVD Tweede Kamer der Staten-Generaal van 29 juni 2006 tot 8 oktober 2010    NT       LE-LH    T1        <NA>       01
NL_Rutte_Mark_1967  pol 28jun2006 14oct2010                             lid Tweede Kamer der Staten-Generaal van 28 juni 2006 tot 14 oktober 2010    NT       LE-LH    T3        <NA>       01
NL_Rutte_Mark_1967  pol 28jun2006 14oct2010                             lid Tweede Kamer der Staten-Generaal van 28 juni 2006 tot 14 oktober 2010    NT       LE-LH    T3        <NA>       01
NL_Rutte_Mark_1967  pol 31may2006      <NA>                                                                 politiek leider VVD vanaf 31 mei 2006    NT       LE-LH T1-FA        <NA>       01
NL_Rutte_Mark_1967 elec 31may2006 22nov2006                   lijsttrekker VVD Tweede Kamerverkiezingen 2006 van 31 mei 2006 tot 22 november 2006    NT       PA-MA    T1        <NA>       01
NL_Rutte_Mark_1967  pol 17jun2004 27jun2006  staatssecretaris van Onderwijs Cultuur en Wetenschap (belast met beroepsonderwijs en volwasseneneduc    NT          EX    T3          NC       01
NL_Rutte_Mark_1967  pol 30jan2003 27may2003                              lid Tweede Kamer der Staten-Generaal van 30 januari 2003 tot 27 mei 2003    NT       LE-LH    T3        <NA>       01
NL_Rutte_Mark_1967  pol 22jul2002 17jun2004  staatssecretaris van Sociale Zaken en Werkgelegenheid (belast met bijstand Wet Sociale Werkvoorzieni    NT          EX    T3          NC       01
NL_Rutte_Mark_1967 prof      2000   feb2002              human-resource manager voor de Raad van Bestuur N.V. Unilever van 2000 tot februari 2002  <NA>        <NA>    T2        1515     <NA>
NL_Rutte_Mark_1967 prof      1997      2000                        personeelsmanager "Van den Bergh Nederland" (Calve) te Delft van 1997 tot 2000  <NA>        <NA>    T2        1515     <NA>
NL_Rutte_Mark_1967  pol      1993      1997                                                                lid hoofdbestuur VVD van 1993 tot 1997    NT       PA-MA    T1        <NA>       03
NL_Rutte_Mark_1967 prof      1992      1997                                                human-resourcemanager N.V. "Unilever" van 1992 tot 1997  <NA>        <NA>    T2        1515     <NA>
NL_Rutte_Mark_1967  pol      1988      1991                       voorzitter JOVD (Jongeren Organisatie Vrijheid en Democratie) van 1988 tot 1991    NT       PA-YO    T1        <NA>       01
NL_Rutte_Mark_1967 prof   feb2002   jul2002 human-resource directeur Unilever werkmaatschappij \\IgloMora\\" te Den Bosch van februari 2002 tot ju  <NA>        <NA>    T2        1515     <NA>
NL_Rutte_Mark_1967  pol   sep2008      <NA>  gastdocent  "Varias College " (Johan de Witt Scholengroep) te `s-Gravenhage vanaf september 2008 (1   <NA>          IG    T3        0600       09
NL_Rutte_Mark_1967  pol        NA      <NA>                                         lid selectiecommissie kandidaten Tweede Kamerfractie VVD 2002    NT       PA-MA T2-23        <NA>       03
NL_Rutte_Mark_1967  pol        NA      <NA>                                                                                 campagneleider VVD 2006    NT       PA-MA T2-22        <NA>       01
```

```
              pers_id type     start       end                                                                                        res_entry_raw level environment  tier policy_area position
NL_Wilders_Geert_1963 elec      2012 12sep2012                                 lijsttrekker PVV Tweede Kamerverkiezingen 2012 tot 12 september 2012    NT       PA-MA    T1        <NA>       01
NL_Wilders_Geert_1963 elec      2010 09jun2010                                       lijsttrekker PVV Tweede Kamerverkiezingen 2010 tot 9 juni 2010    NT       PA-MA    T1        <NA>       01
NL_Wilders_Geert_1963  pol 11mar2010 01jul2010                                 lid gemeenteraad van `s-Gravenhage van 11 maart 2010 tot 1 juli 2010    MU          LE    T3        <NA>       01
NL_Wilders_Geert_1963  pol 23nov2006   aug2012                        fractievoorzitter PVV Tweede Kamer der Staten-Generaal vanaf 23 november 2006    NT       LE-LH    T1        <NA>       01
NL_Wilders_Geert_1963 elec 20sep2006 22nov2006 lijsttrekker PVV/Groep-Wilders Tweede Kamerverkiezingen 2006 van 20 september 2006 tot 22 november 2    NT       PA-MA    T1        <NA>       01
NL_Wilders_Geert_1963  pol 22feb2006      <NA>                                                           politiek leider PVV vanaf 22 februari 2006    NT       LE-LH T1-FA        <NA>       01
NL_Wilders_Geert_1963  pol 15jan2006      <NA>       voorzitter Stichting Ondersteuning Tweede Kamerfractie PVV/Groep-Wilders vanaf 15 januari 2006    NT       PA-MA    T2          NC       01
NL_Wilders_Geert_1963  pol 30mar2005      <NA>                   lid bestuur Vereniging PVV/Groep-Wilders vanaf 30 maart 2005 (enige (bestuurs)lid)    NT       PA-MA    T1        <NA>       03
NL_Wilders_Geert_1963  pol 24nov2004      <NA>                   lid bestuur Stichting PVV/Groep-Wilders vanaf 24 november 2004 (enige bestuurslid)    NT       PA-MA    T1        <NA>       03
NL_Wilders_Geert_1963  pol 02sep2004 23nov2006 fractievoorzitter Groep-Wilders Tweede Kamer der Staten-Generaal van 2 september 2004 tot 23 novembe    NT       LE-LH    T1        <NA>       01
NL_Wilders_Geert_1963  pol 26jul2002   aug2012                                              lid Tweede Kamer der Staten-Generaal vanaf 26 juli 2002    NT       LE-LH    T3        <NA>       01
NL_Wilders_Geert_1963  pol 26jul2002      <NA>                                              lid Tweede Kamer der Staten-Generaal vanaf 26 juli 2002    NT       LE-LH    T3        <NA>       01
NL_Wilders_Geert_1963  pol 25aug1998 22may2002                            lid Tweede Kamer der Staten-Generaal van 25 augustus 1998 tot 23 mei 2002    NT       LE-LH    T3        <NA>       01
NL_Wilders_Geert_1963  pol      1998      2002                                        lid Comite democratie en mensenrechten Iran van 1998 tot 2002  <NA>          IG    T1   0200;2300       03
NL_Wilders_Geert_1963  pol 01oct1997   apr1998                                       lid gemeenteraad van Utrecht van 1 oktober 1997 tot april 1998    MU          LE    T3        <NA>       01
NL_Wilders_Geert_1963  pol      1990   aug1998 beleidsmedewerker sociale zaken en sociaal-economisch beleid en speechschrijver VVD-fractie Tweede K    NT       LE-LH T3-FA        <NA>       07
NL_Wilders_Geert_1963 prof      1986      1988                            wetstechnisch medewerker SVR (Sociale Verzekeringsraad) van 1986 tot 1988  <NA>        <NA>  <NA>        1517     <NA>
NL_Wilders_Geert_1963 prof      1984      1986                                      medewerker afdeling Verdragen Ziekenfondsraad van 1984 tot 1986  <NA>        <NA>  <NA>        1517     <NA>
NL_Wilders_Geert_1963  pol   feb2010      <NA>                                 eigenaar en enig aandeelhouder  "OnLiberty" B.V. vanaf februari 2010  <NA>          IG    T3        1515       09
NL_Wilders_Geert_1963  pol        NA      <NA>                                                              lid Parlementaire Assemblee van de NAVO    IN          LE    NC          NC       03
NL_Wilders_Geert_1963  pol        NA      <NA>                                lid bestuur Kappeyne van de Coppello Stichting omstreeks januari 2000    NT       PA-MA T2-40        <NA>       03
```

Other paper specific data modules

There are three additional ‘modules’ (dataframes that are not part of the core data-set but nevertheless compatible with it). Firstly, there is a module of ministerial episodes (’MINE’) that contains information on who was a minister in which parliament. There is a module with gender quota data (’QUOT’) that contains information on which party had voluntary gender quotas, with start and end dates. Finally, ‘FACT’ contains information on how many seats each party held in which parliament. The exact source of each data element is specified in the codebook in the appendix to this thesis.

Conclusion

This chapter has several implications for political career researchers. First, it suggests a general data setup that can be used to flexibly store political career data. Second, it shows how this information can be stored in a relational database structure and can be extracted with the ‘sqldf’ package in R. All the empirical chapters that follow use this database. All the data that is used in the presented analysis has been generated with SQL queries very similar to the ones just presented.

A whole universe of data-structures and research questions can be answered with this data. Despite my best efforts, I have left a vast amount of them untouched. Additional work on those questions will reveal further directions this database should take. As long as such work contributes to a central shared data structure such as the one presented, the data collection effort that goes into these projects will be accumulative. Given the limited resources available for social science research, a shared attempt to collect and store political career data might prove to be crucial to generating an increased understanding of political career dynamics and the scientific and societal consequences that will follow from the increased insight that this endeavour provides.

References

Döring, H. and Manow, P. (2019). Parliaments and governments database (ParlGov): Information on parties, elections and cabinets in modern democracies. Development version.

Hernandez, M.J. (2013). Database Design for Mere Mortals: A Hands-on Guide to Relational Database Design. Pearson Education

Krause, W., Lehmann, P., Lewandowski, J., Matthieß, T., Merz, N., Regel, S., Werner, A. (2020): Manifesto Corpus. Berlin: WZB Berlin Social Science Center.

Louwerse, T., Otjes, S., van Vonno, C. (2018). The Dutch parliamentary behaviour dataset. *Acta Politica*, 53(1), 149-166.