Bavarian Election System

Stefan Kussmaul & Vlad Kolesnykov

Technologies

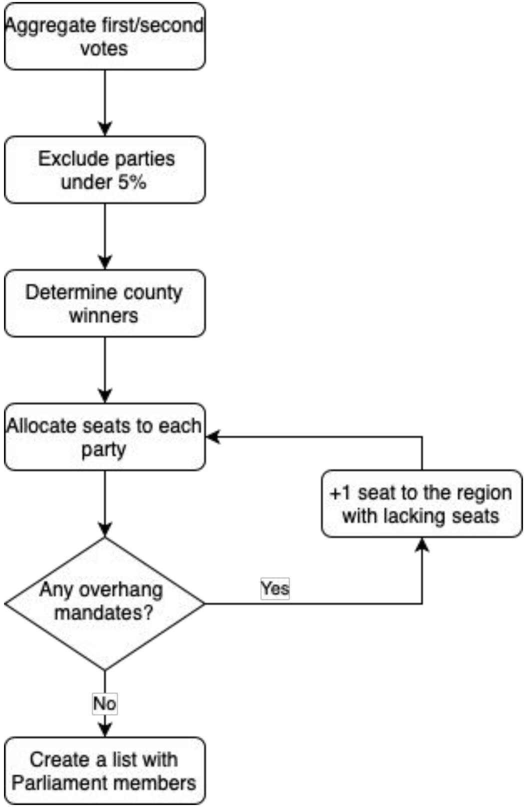
Backend: Flask + PostgreSQL

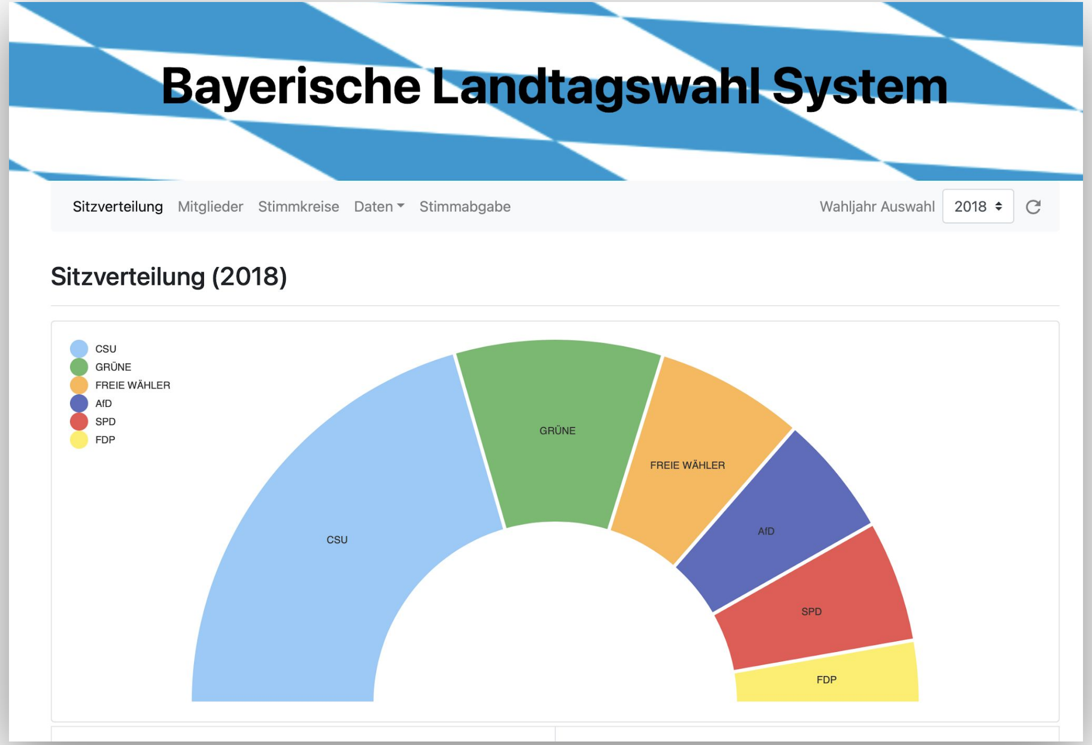
Data Generator: Python

Frontend: React + Bootstrap

Demonstration

Distribution of seats in the Parliament


Aggregate first/second votes
Exclude parties under 5%
Determine county winners
Allocate seats to each party
Any overhang mandates?
Yes
+1 seat to the region with lacking seats
No
Create a list with Parliament members



Sitzverteilung (2018)

Performance

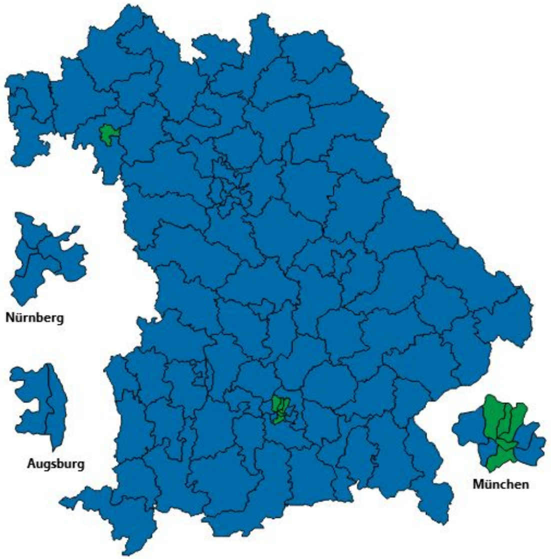
	1 user 1 sec wait n=460	50 users 5 sec wait n=2200	500 users 5 sec wait n=20,000	250 users 1 sec wait n=35,000	500 users 1 sec wait n=50,000
Avg. RPS	1.0	9.6	98.9	160.9	186.2
Failure Rate	0.0%	0.001%*	0.002%*	0.003%*	14.6%
Q1	5 ms	9 ms	32 ms	530 ms	1630 ms
Q2	13 ms	20 ms	42 ms	539 ms	1638 ms
Q3	5 ms	10 ms	37 ms	535 ms	1638 ms
Q4	7 ms	12 ms	37 ms	533 ms	1652 ms
Q5	10 ms	11 ms	36 ms	533 ms	1619 ms
Q6	8 ms	10 ms	33 ms	530 ms	1619 ms

*All failures due to invalid URL during benchmarking

Tested on Intel Core i5-7300HQ CPU @ 2.50 GHz, 16GB RAM. Local Flask development server

Ideas for Extension

- Performance Improvements
  - Deploy to uWSGI server
  - Change to a faster framework (e.g. FastAPI or Falcon)
- Interactive Stimmkreis map
- Authenticated dashboard for Voterkey-registration



Interactive map from https://www.landtagswahl2018.bayern.de/