KOALA: A new paradigm for election coverage

An opinion poll based “now-cast” of probabilities of events in multi-party electoral systems

Alexander Bauer

Statistical Consulting Unit StaBLab, LMU Munich

DAGStat | March 20, 2019 | Munich

KOALA

KoalitionsAnalyse

Alexander Bauer

Statistical Consulting Unit StaBLab, LMU Munich

DAGStat | March 20, 2019 | Munich

KOALA

KoalitionsAnalyse

Collaborators

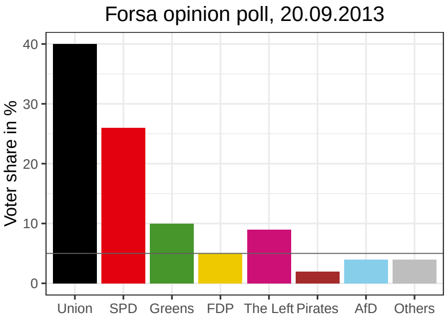
Dr. Andreas Bender	Nuffield Department of Clinical Medicine, University of Oxford, United Kingdom
Dr. André Klima	StaBLab, LMU Munich
Prof. Dr. Helmut Küchenhoff	StaBLab, LMU Munich

Outline

1. Motivation
2. Methods
3. Technical implementation
4. Conclusion

Outline

❶ Motivation



Questions of interest

- Which parties will pass the 5% hurdle and enter the parliament?
- Which parties will form the governing coalition?
- Which party will have the third largest share of votes?

❶ Motivation

Reported voter shares

Union	SPD	Greens	FDP	The Left	Pirates	AfD	Others
40%	26%	10%	5%	9%	2%	4%	5%

Redistributed voter shares (based on 5% hurdle)

Union	SPD	Greens	FDP	The Left	Pirates	AfD	Others
44.44%	28.89%	11.11%	5.56%	10.00%	-	-	-

- Union-FDP have a joint seat share of exactly 50%
- Stating that Union-FDP would thus miss a joint majority would neglect sample uncertainty

⇒ We calculate event probabilities that fully reflect sample uncertainty

❶ Motivation

Reported voter shares

Union	SPD	Greens	FDP	The Left	Pirates	AfD	Others
40%	26%	10%	5%	9%	2%	4%	5%

Redistributed voter shares (based on 5% hurdle)

Union	SPD	Greens	FDP	The Left	Pirates	AfD	Others
44.44%	28.89%	11.11%	5.56%	10.00%	-	-	-

- Union-FDP have a joint seat share of exactly 50%
- Stating that Union-FDP would thus miss a joint majority would neglect sample uncertainty

⇒ We calculate event probabilities that fully reflect sample uncertainty

❶ Motivation

We aim to do now-casting

- We incorporate the uncertainty as reported by the polling agencies
- Potential house biases or an industry bias are not accounted for

We do not aim to do for-casting

- Our approach simply communicates sample uncertainty in a novel way
- Also, a relevant share of voters is still undecided shortly before election day (Küchenhoff et al., 2018)

❶ Motivation

We aim to do now-casting

- We incorporate the uncertainty as reported by the polling agencies
- Potential house biases or an industry bias are not accounted for

We do not aim to do for-casting

- Our approach simply communicates sample uncertainty in a novel way
- Also, a relevant share of voters is still undecided shortly before election day (Küchenhoff et al., 2018)

Outline

❷ Methods

Estimating probabilities of events (POEs)

Given one opinion poll with sample size n:

$$\boldsymbol{X} = (X_1, \dots, X_P)^T \sim Multinomial(n, \theta_1, \dots, \theta_P),$$

with voter counts X_j and the true percentage of voters θ_j per party j

Using an **uninformative Dirichlet prior** (Gelman et al., 2013)

$$\boldsymbol{\theta} = (\theta_1, \dots, \theta_P)^T \sim Dirichlet(\alpha_1, \dots, \alpha_P),$$

$$\text{with} \qquad \alpha_1 = \dots = \alpha_P = \frac{1}{2},$$

a **Dirichlet posterior distribution** results for $\boldsymbol{\theta}|\boldsymbol{x}$:

$$\boldsymbol{\theta}|\mathbf{x} \sim Dirichlet(x_1 + 1/2, \dots, x_P + 1/2).$$

❷ Methods

Estimating probabilities of events (POEs)

Given one opinion poll with sample size n:

$$\boldsymbol{X} = (X_1, \ldots, X_P)^T \sim \mathit{Multinomial}(n, \theta_1, \ldots, \theta_P),$$

with voter counts X_j and the true percentage of voters θ_j per party j

Using an **uninformative Dirichlet prior** (Gelman et al., 2013)

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)^T \sim \mathit{Dirichlet}(\alpha_1, \ldots, \alpha_P),$$

with $$\alpha_1 = \ldots = \alpha_P = \frac{1}{2},$$

a **Dirichlet posterior distribution** results for $\boldsymbol{\theta}|\boldsymbol{x}$:

$$\boldsymbol{\theta}|\mathbf{x} \sim \mathit{Dirichlet}(x_1 + 1/2, \ldots, x_P + 1/2).$$

❷ Methods

Estimating probabilities of events (POEs)

Given the **posterior distribution of voter shares** we can use **Monte Carlo simulations** to estimate POEs:

1. Simulate 10 000 election outcomes from the posterior
2. If necessary: Redistribute voter shares to get obtained seats in parliament
3. $\text{POE} = \frac{\#\text{event}}{\text{number of simulations}}$

Example

Given the Forsa poll, the coalition of Union-FDP obtained a majority of seats in 2 633 of 10 000 simulations

$\Rightarrow$ POE $\approx 26\%$

❷ Methods

Estimating probabilities of events (POEs)

Given the **posterior distribution of voter shares** we can use
Monte Carlo simulations to estimate POEs:

1. Simulate 10 000 election outcomes from the posterior
2. If necessary: Redistribute voter shares to get obtained seats in parliament
3. $\text{POE} = \frac{\#\text{event}}{\text{number of simulations}}$

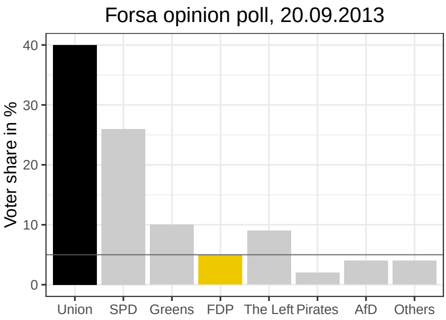
Example

Given the Forsa poll, the coalition of Union-FDP obtained a majority of seats in 2 633 of 10 000 simulations

$\Rightarrow \text{POE} \approx 26\%$

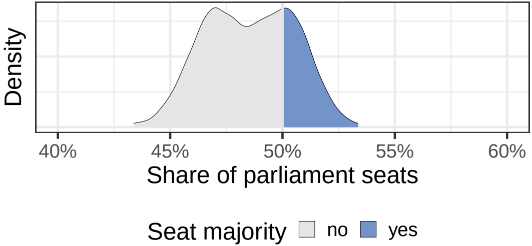
❷ Methods

Voter shares



❷ Methods

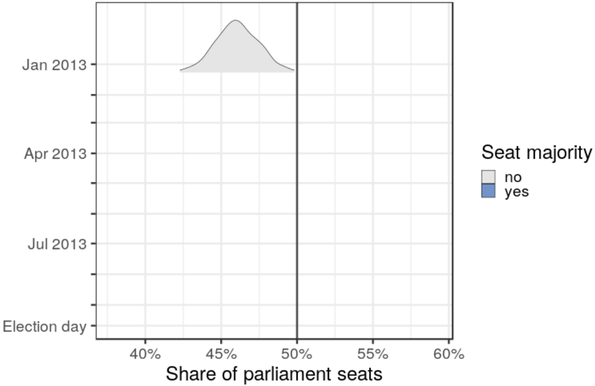
Posterior distribution of joint CDU-FDP seat share



$\Rightarrow$ POE $\approx 26\%$

❷ Methods

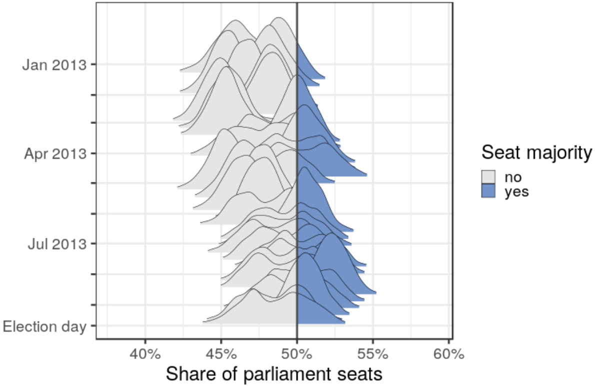
Visualization using ridgeline plots (Wilke, 2017)



I'm a .gif, click me (in Adobe Acrobat)!

❷ Methods

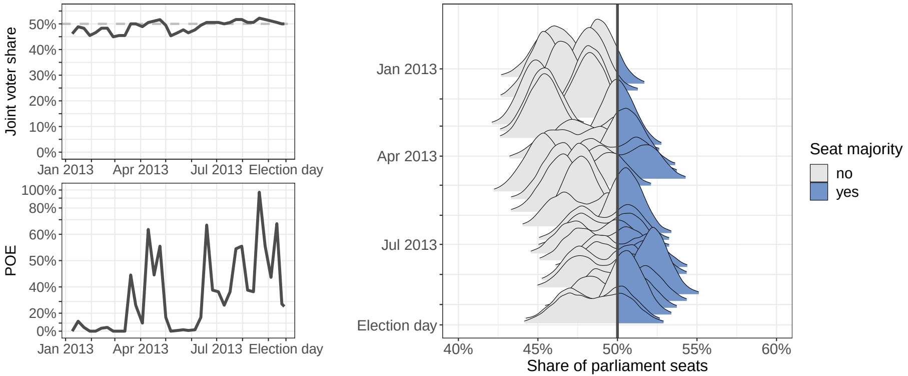
Visualization using ridgeline plots (Wilke, 2017)



I'm a .gif, click me (in Adobe Acrobat)!

❷ Methods

Visualization using ridgeline plots (Wilke, 2017)



❷ Methods

Pooling

We aggregate multiple polls to reduce sample uncertainty.
In case of multiple random samples:

$$\left(\sum_i X_{i1}, \ldots, \sum_i X_{iP}\right)^T \sim Multinomial\left(\sum_i n_i, \theta_1, \ldots, \theta_P\right).$$

We account for correlations between polling agencies by using an **effective sample size** (Hanley et al., 2003).

⇒ **Example:** Pooling two polls with 1 500 and 2 000 respondents (where the strongest party obtained 40%), we get a conservative effective sample size of $n_{\text{eff}} = 2\,341$.

❷ Methods

Pooling

We aggregate multiple polls to reduce sample uncertainty.
In case of multiple random samples:

$$\left(\sum_i X_{i1}, \ldots, \sum_i X_{iP}\right)^T \sim \mathit{Multinomial}\left(\sum_i n_i, \theta_1, \ldots, \theta_P\right).$$

We account for correlations between polling agencies by using an **effective sample size** (Hanley et al., 2003).

⇒ **Example:** Pooling two polls with 1 500 and 2 000 respondents
(where the strongest party obtained 40%),
we get a conservative effective sample size of $n_{\text{eff}} = 2\,341$.

❷ Methods

Pooling in practice

- We only pool surveys published in the last 14 days
- We only include one survey per polling agency

Correction of rounding errors

Party shares are only published with a certain accuracy.
We add **uniformly distributed random noise** to avoid potential biases.

❷ Methods

Pooling in practice

- We only pool surveys published in the last 14 days
- We only include one survey per polling agency

Correction of rounding errors

Party shares are only published with a certain accuracy.
We add **uniformly distributed random noise** to avoid potential biases.

Outline

1. Motivation
2. Methods
3. Technical implementation
4. Conclusion

❷ Technical implementation

R package `coalitions`

Functionality

- Scrapes wahlrecht.de for (new) polls
- Calculate pooled sample
- Calculate and sample from posterior distribution
- Redistribute votes below 5% threshold
  and calculate parliament seats (e.g. based on method by Sainte-Lague-Scheppers)
- Calculate coalition probabilities

More on github.com/adibender/coalitions

❷ Technical implementation

R package `coalitions`

```
> library(coalitions)
> library(tidyverse)

> surveys <- get_surveys()
> surveys

# A tibble: 7 x 2
  pollster   surveys
  <chr>      <list>
1 allensbach <tibble [42 x 5]>
2 emnid      <tibble [226 x 5]>
3 forsa      <tibble [236 x 5]>
...

> survey <- surveys %>% unnest() %>% slice(1)
> survey %>% get_probabilities(list(c("cdu","fdp")), nsim = 10000) %>%
  unnest()
# A tibble: 1 x 4
  pollster   date       coalition probability
  <chr>      <date>     <chr>           <dbl>
1 allensbach 2019-02-19 cdu_fdp             0
```

Web-Interface

Communicating the results

1. Website koala.stat.uni-muenchen.de
   ⇒ Automatic updates scraping data from wahlrecht.de
2. Twitter @KOALA_LMU
   ⇒ Automatic tweets of new results
3. Blog koala-blog.netlify.com

Technical implementation in R

- User interface was built with the shiny package
- Server is based on Shiny Server Open Source
- Tweets are sent with the twitteR package

Web-Interface

Communicating the results

1. Website `koala.stat.uni-muenchen.de`

   ⇒ Automatic updates scraping data from wahlrecht.de
2. Twitter @KOALA_LMU

   ⇒ Automatic tweets of new results
3. Blog koala-blog.netlify.com

Technical implementation in R

- User interface was built with the shiny package
- Server is based on Shiny Server Open Source
- Tweets are sent with the twitteR package

Outline

1. Motivation

2. Methods

3. Technical implementation

4. Conclusion

❺ Conclusion

KOALA

KoalitionsAnalyse

The KOALA approach

- New paradigm for opinion poll coverage
- Bayesian approach to now-cast POEs
- Sample uncertainty is reduced by pooling multiple polls
- Communication to the general public

- **Keep in mind:** We do not make predictions

References

KOALA

Bauer, A., Bender, A., Klima, A., and Küchenhoff, H. (2018) KOALA: A new paradigm for election coverage. *arXiv.org*. URL `http://arxiv.org/abs/1807.09665`.

Bender, A., and Bauer, A. (2018) coalitions: Coalition probabilities in multi-party democracies. *The Journal of Open Source Software*. doi: 10.21105/joss.00606.

URL `http://joss.theoj.org/papers/10.21105/joss.00606`.

Methods

Gelman, A. et al. (2013) *Bayesian data analysis*, volume 3. CRC press Boca Raton, FL.

Hanley, J. A. et al. (2003) Statistical analysis of correlated data using generalized estimating equations: an orientation. *American journal of epidemiology*, 157(4), 364–375.

Küchenhoff, H. et al. (2018) *Universitätsstudie zur Bayernwahl USBW 18 (München – Passau – Regensburg). Erste Ergebnisse – Oktober 2018.*

URL `https://www.stablab.stat.uni-muenchen.de/lehre/pdfs/usbw18.pdf`.

Further software

Chang, W. et al. (2017) *shiny: Web Application Framework for R*.
URL `https://CRAN.R-project.org/package=shiny`. R package version 1.0.5.

Wilke, C. O. (2017) *ggridges: Ridgeline Plots in 'ggplot2'*.
URL `https://CRAN.R-project.org/package=ggridges`. R package version 0.4.1.