Running the main polarization estimation codes in **/cluster/analysis_dates/qh_analysis/** folder

- We have used a computing cluster to estimate these scripts
- The important scripts are the `tool-exec` bash scripts.
- Each of these tool-exec scripts and the scripts they run are located in separate folders
- Scripts run on `partyvar` level. For example, if you want to run the main left-right comparison, you set `partyvar='left'`.
- Scripts create a working directory called [partyvar] within the `temp` folder, into which the intermediate files are being created. Results are similarly output under `analysis_dates/output/[partyvar]` folder.
- **All final plots and tables are created in a different place (see /final_estimation/ folder). Those final estimations depend on results estimated using scripts in /cluster/analysis_dates/ folder. We have stored the estimation results in the folder, so it is possible to run also these final plots without first running scripts in /cluster/analysis_dates/ folder**
- We provide some additional short README files containing relevant information in different folders.

Parameter	Description	Value	Description
`partyvar`	The party split used for the analysis. `partyvar='left'` runs the main left-right speech comparison.	`left`	left-right comparison
		`govparty`	government-opposition comparison
		`leftnonsmp`	left-right comparison without SMP
		`leftnonvennamo`	left-right comparison without Vennamo
		`demkesk`	SDP-Centre Party comparison
		`demkok`	SDP-Coalition Party comparison
		`kokkesk`	Coalition Party-Centre Party comparison
		`vasdem`	Vas-SDP comparison
		`vaskesk`	Vas-Centre Party comparison
		`vaskok`	Vas-Coalition Party comparison

Parameter	Description	Value	Description
`nr_datasets`	into how many datasets data is split	10 by default	
`Cpar`	determines which counts dataset is used	`Cadj` (default)	`speaker_phrase_counts_bipartisan_adj.rds` (more procedural phrases removed)
		empty	`speaker_phrase_counts_bipartisan.rds`
`covariates`		`c0`	No covariates
		`c1`	controls for party's government-opposition status
		`c2`	controls for gov-opp status + dialect
		`c3`	controls for gov-opp status + dialect + gender
		`c4`	controls for dialect + gender
		`c99`	no controls, try a low value for the penalty coefficient psi ψ
		`c100`	no controls, try different values for the penalty coefficient psi ψ
`penalty`	whether or not LASSO is used		
`fake_indicator`	placebo or not placebo		

`tool-exec-1-prepare-data.sh`

Description: Runs three different scripts that prepare covariates and split the datasets for the purposes of the main analyses.

`prepare-data-1.py`:

Description: Create party indicators and dialect region variable; only keep the MPs that served in the parliament during the parliamentary year. Create `id` variable, which concatenates year and `speaker_id`. Note that `speaker_id` is not used for anything as such

Input: `cluster/build_dates/output/mps-ministers.csv`

Output: `cluster/analysis_dates/input/speaker_metadata_bipartisan.csv`

prepare-data-2.R:

Description: Do some small changes in data to better fit R way of doing things: make categorical variables factor variables. Drop MPs from Åland. Drop speech by parliament chair.
Create `randlabel` variable, which is a randomly created party variable for the placebo tests.

Input:

- `cluster/analysis_dates/input/bow-qh-1907-2018-tf-25-df-5-ytf-5.csv`
- `cluster/analysis_dates/input/speaker_metadata_bipartisan.csv`

Output:

- `cluster/analysis_dates/temp/[partyvar]/speaker_metadata_bipartisan.rds`: speaker metadata and the specified party indicator
- `cluster/analysis_dates/temp/[partyvar]/speaker_phrase_counts_bipartisan.rds`: matrix with phrase counts only; speaker_id-year id as a row name

prepare-data-2b.R:

Description:

Filter out some more procedural language.

Input:

- `cluster/analysis/input/list-adjust-c2.csv`
- `cluster/analysis/temp_qh/[partyvar]/speaker_phrase_counts_bipartisan.rds`

Output:

- `cluster/analysis/temp_qh/[partyvar]/speaker_phrase_counts_bipartisan_adj.rds`

prepare-data-3.R:

Description:

Split data (both metadata and phrase count file) into [nr_datasets] different datasets (default = 10). There were some memory-related issues why I ended up using a split.

Add μ (nr of phrases spoken by a given MP during the year) into data

Input:

- `cluster/analysis/temp_qh/[partyvar]/speaker_phrase_counts_bipartisan_adj.rds`
- `cluster/analysis/temp_qh/[partyvar]/speaker_metadata_bipartisan.rds`

Output:

- `cluster/analysis/temp_qh/[partyvar]/speaker_phrase_counts_bipartisan_adj_1.rds`

- `cluster/analysis/temp_qh/[partyvar]/speaker_phrase_counts_bipartisan_adj_2.rds`, etc.
- `cluster/analysis/temp_qh/[partyvar]/speaker_metadata_data_1.rds`, etc.

tool-exec-2-estimate.sh

Description:

Chooses the main analysis program based on the covariates specified and runs it. If `randlabel=1`, computes the placebo series.

Input

Output:

- `cluster/analysis/temp_qh/[partyvar]/utility_dem_[Cpar]_[dataset_nr].rds`
- `cluster/analysis/temp_qh/[partyvar]/utility_rep_[Cpar]_[dataset_nr].rds`
- `cluster/analysis/temp_qh/[partyvar]/utility_[Cpar]_[dataset_nr].rds`
- `cluster/analysis/temp_qh/[partyvar]/phi_[Cpar]_[dataset_nr].rds`

tool-exec-3-combine.sh

Description:

Combine data from the `nr_datasets` datasets.

tool-exec-4a-subsample.sh

Description:

Estimates the partisanship series 100 times for a subsample of the data. A different program is executed depending on which covariates are chosen. Subsample is drawn randomly without replacement. Subsample size needs to be changed manually in the code, i.e., it is hardcoded in the subprogram but could well be improved by writing subsampling percentage into a variable.

tool-exec-5-cis.sh

Description:

Construct confidence intervals based on the distribution computed in 4a