isq_wang_xu_output

Guan Wang

2025-03-17

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          4
## minor          1.3
## year           2022
## month          03
## day            10
## svn rev        81868
## language       R
## version.string R version 4.1.3 (2022-03-10)
## nickname       One Push-Up
```

Install packages needed for the analysis

```
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
packageVersion("stargazer") #'5.2.3'
```

```
## [1] '5.2.3'
```

```
#install.packages("lme4")
library(lme4)
```

```
## Loading required package: Matrix
```

```r
packageVersion("lme4") #‘1.1.31’
```

```
## [1] ’1.1.31’
```

```r
#install.packages("boot")
library(boot)
packageVersion("boot") #‘1.3.28’
```

```
## [1] ’1.3.28’
```

```r
library(ggplot2)
packageVersion("ggplot2") #‘3.4.0’
```

```
## [1] ’3.4.0’
```

```r
#install.packages("writexl")
library(writexl)
packageVersion("writexl") #‘1.4.1’
```

```
## [1] ’1.4.1’
```

```r
#install.packages("multilevel")
library(multilevel)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: ’nlme’
```

```
## The following object is masked from ’package:lme4’:
##
##     lmList
```

```
## Loading required package: MASS
```

```r
packageVersion("multilevel") #‘2.7’
```

```
## [1] ’2.7’
```

```r
#install.packages("nlme")
library(nlme)
packageVersion("nlme") #‘3.1.161’
```

```
## [1] ’3.1.161’
```

```r
#install.packages("dfoptim")
library(dfoptim)
packageVersion("dfoptim") #‘2020.10.1’
```

```
## [1] ’2020.10.1’
```

```r
#install.packages("optimx")
library(optimx)
```

```
##
## Attaching package: ’optimx’

## The following object is masked from ’package:nlme’:
##
##     coef<-
```

```r
packageVersion("optimx") #‘2022.4.30’
```

```
## [1] ’2022.4.30’
```

```r
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following object is masked from ’package:MASS’:
##
##     select

## The following object is masked from ’package:nlme’:
##
##     collapse

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```r
packageVersion("dplyr") #‘1.0.10’
```

```
## [1] ’1.0.10’
```

```r
#install.packages("afex")
library(afex)
```

```
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/

## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'S', 'KR', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
## - NEWS: emmeans() for ANOVA models now uses model = 'multivariate' as default.
## - Get and set global package options with: afex_options()
## - Set orthogonal sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************

##
## Attaching package: 'afex'

## The following object is masked from 'package:lme4':
##
##     lmer
```

```r
packageVersion("afex") #'1.2.0'
```

```
## [1] '1.2.0'
```

```r
#install.packages("vcd")
library(vcd)
```

```
## Loading required package: grid
```

```r
packageVersion("vcd") #'1.4.10'
```

```
## [1] '1.4.10'
```

```r
#install.packages("plyr")
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
packageVersion("plyr") #'1.8.8'
```

```
## [1] '1.8.8'
```

```
#install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v purrr   1.0.1
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x plyr::arrange()   masks dplyr::arrange()
## x dplyr::collapse() masks nlme::collapse()
## x purrr::compact()  masks plyr::compact()
## x plyr::count()     masks dplyr::count()
## x tidyr::expand()   masks Matrix::expand()
## x plyr::failwith()  masks dplyr::failwith()
## x dplyr::filter()   masks stats::filter()
## x plyr::id()        masks dplyr::id()
## x dplyr::lag()      masks stats::lag()
## x plyr::mutate()    masks dplyr::mutate()
## x tidyr::pack()     masks Matrix::pack()
## x plyr::rename()    masks dplyr::rename()
## x dplyr::select()   masks MASS::select()
## x plyr::summarise() masks dplyr::summarise()
## x plyr::summarize() masks dplyr::summarize()
## x tidyr::unpack()   masks Matrix::unpack()
```

```
packageVersion("tidyverse") #'1.3.2'
```

```
## [1] '1.3.2'
```

```
#install.packages("magrittr")
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
packageVersion("magrittr") #'2.0.3'
```

```
## [1] '2.0.3'
```

```
#install.packages("partykit")
library(partykit)
```

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

```
packageVersion("partykit") #'1.2.16'
```

```
## [1] '1.2.16'
```

```
#install.packages("pdp")
library(pdp)
```

```
##
## Attaching package: 'pdp'
```

```
## The following object is masked from 'package:purrr':
##
##     partial
```

```
packageVersion("pdp") #'0.8.1'
```

```
## [1] '0.8.1'
```

```
#install.packages("iml")
library(iml)
packageVersion("iml") #'0.11.1'
```

```
## [1] '0.11.1'
```

```
#install.packages("DALEX")
library(DALEX)
```

```
## Welcome to DALEX (version: 2.4.2).
## Find examples and detailed introduction at: http://ema.drwhy.ai/
```

```
##
## Attaching package: 'DALEX'
```

```
## The following object is masked from 'package:dplyr':
##
##     explain
```

```
packageVersion("DALEX") #‘2.4.2’
```

```
## [1] ’2.4.2’
```

```
#install.packages("relaimpo")
library(relaimpo)
```

```
## Loading required package: survey
```

```
## Loading required package: survival
```

```
##
## Attaching package: ’survival’
```

```
## The following object is masked from ’package:boot’:
##
##     aml
```

```
##
## Attaching package: ’survey’
```

```
## The following object is masked from ’package:graphics’:
##
##     dotchart
```

```
## Loading required package: mitools
```

```
## This is the global version of package relaimpo.
```

```
## If you are a non-US user, a version with the interesting additional metric pmvd is available
```

```
## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```
packageVersion("relaimpo") #‘2.2.6’
```

```
## [1] ’2.2.6’
```

```
#install.packages("mitml")
library(mitml)
```

```
## *** This is beta software. Please report any bugs!
## *** See the NEWS file for recent changes.
```

```
packageVersion("mitml") #‘0.4.3’
```

```
## [1] ’0.4.3’
```

```r
#install.packages("rr2")
library(rr2)
```

```
##
## Attaching package: 'rr2'

## The following object is masked from 'package:boot':
##
##     inv.logit
```

```r
packageVersion("rr2") #'1.1.0'
```

```
## [1] '1.1.0'
```

```r
#install.packages("r2mlm")
library(r2mlm)
packageVersion("relaimpo") #'2.2.6'
```

```
## [1] '2.2.6'
```

```r
#install.packages("AER")
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:purrr':
##
##     some

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:boot':
##
##     logit

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich
```

```
packageVersion("AER") #‘1.2.10’
```

```
## [1] '1.2.10'
```

```
#install.packages("BayesPostEst")
library(BayesPostEst)
packageVersion("BayesPostEst") #‘0.3.2’
```

```
## [1] '0.3.2'
```

```
#install.packages("rstanarm")
library(rstanarm)
```

```
## Loading required package: Rcpp

## This is rstanarm version 2.21.4

## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

##   options(mc.cores = parallel::detectCores())

##
## Attaching package: 'rstanarm'

## The following object is masked from 'package:car':
##
##     logit

## The following object is masked from 'package:rr2':
##
##     R2

## The following object is masked from 'package:boot':
##
##     logit
```

```
packageVersion("rstanarm") #‘2.21.4’
```

```
## [1] ’2.21.4’
```

```
#install.packages("devtools")
library(devtools)
```

```
## Loading required package: usethis
```

```
packageVersion("devtools") #‘2.4.4’
```

```
## [1] ’2.4.4’
```

```
#install.packages("foreign")
library(foreign)
packageVersion("foreign") #‘0.8.84’
```

```
## [1] ’0.8.84’
```

```
#install.packages("mvtnorm")
library(mvtnorm)
packageVersion("mvtnorm") #‘1.1.3’
```

```
## [1] ’1.1.3’
```

```
#install.packages("interplot")
library(interplot)
```

```
## Loading required package: abind

## Loading required package: arm

##
## arm (Version 1.13-1, built: 2022-8-25)

## Working directory is C:/isq

##
## Attaching package: ’arm’

## The following objects are masked from ’package:rstanarm’:
##
##     invlogit, logit

## The following object is masked from ’package:car’:
##
##     logit

## The following object is masked from ’package:boot’:
##
##     logit
```

```
packageVersion("interplot") #'0.2.3'
```

```
## [1] '0.2.3'
```

```
#install.packages("ggforce")
library(ggforce)
packageVersion("ggforce") #'0.4.1'
```

```
## [1] '0.4.1'
```

Load the data

```
data1<-read.csv("C://isq//china bills 2009-2022.csv",fileEncoding="UTF-8-BOM")
```

Change the class of variables

```
data1$dw_nominate_dim1<-as.numeric(data1$dw_nominate_dim1)
```

```
## Warning: NAs introduced by coercion
```

```
data1$pages<-as.numeric(data1$pages)
```

```
## Warning: NAs introduced by coercion
```

```
data1$Legislation.type<-ifelse(data1$Legislation.type=="Bill","Bill","Resolution")
data1$term_has_served<-as.numeric(data1$term_has_served)
data1$month_since_beginning_congress<-as.numeric(data1$month_since_beginning_congress)
```

##Select legislation that describe China as a threat

```
data11<-subset(data1,data1$anti.china==1)
nrow(data11)
```

```
## [1] 603
```

Create a new variable that counts the number of China-related legislative proposals introduced in each congressional session

```
table(data11$Congress)
```

```
##
## 111th Congress (2009-2010) 112th Congress (2011-2012)
##                         38                         36
## 113th Congress (2013-2014) 114th Congress (2015-2016)
##                         30                         23
## 115th Congress (2017-2018) 116th Congress (2019-2020)
##                         49                        166
## 117th Congress (2021-2022)
##                        261
```

```
number_of_obs <- ifelse(data11$Congress == "111th Congress (2009-2010)", 38,
                        ifelse(data11$Congress == "112th Congress (2011-2012)", 36,
                               ifelse(data11$Congress == "113th Congress (2013-2014)", 30,
                                      ifelse(data11$Congress == "114th Congress (2015-2016)", 23,
                                             ifelse(data11$Congress == "115th Congress (2017-2018)", 49
                                                    ifelse(data11$Congress == "116th Congress (2019-202
                                                           ifelse(data11$Congress == "117th Congress (2
```

```
data11$number_of_obs <- as.numeric(number_of_obs)
```

Table 2

Sponsor's ideology

```
subset_data <- data11[!is.na(data11$dw_nominate_dim1),]

mean(abs(subset_data$dw_nominate_dim1));sd(abs(subset_data$dw_nominate_dim1));min(abs(subset_data$dw_nor
```

```
## [1] 0.4726483
```

```
## [1] 0.1685975
```

```
## [1] 0.088
```

```
## [1] 0.936
```

Sponsors' years of service

```
mean(subset_data$term_has_served);sd(subset_data$term_has_served);min(subset_data$term_has_served);max(s
```

```
## [1] 10.69167
```

```
## [1] 10.43686
```

```
## [1] 0
```

```
## [1] 40
```

Legislative length

```
mean(log(subset_data$pages));sd(log(subset_data$pages));min(log(subset_data$pages));max(log(subset_data$
```

```
## [1] 1.860147
```

```
## [1] 0.9056797
```

```
## [1] 0
```

```
## [1] 7.751905
```

Resolution

```
subset_data$Legislation.type<-ifelse(subset_data$Legislation.type=="Resolution",1,0)
mean(subset_data$Legislation.type);sd(subset_data$Legislation.type);min(subset_data$Legislation.type);ma
```

```
## [1] 0.2333333
```

```
## [1] 0.4233055
```

```
## [1] 0
```

```
## [1] 1
```

###Months have passed

```
mean(subset_data$month_since_beginning_congress);sd(subset_data$month_since_beginning_congress);min(subs
```

```
## [1] 11.24833
```

```
## [1] 6.809107
```

```
## [1] 1
```

```
## [1] 24
```

Salience

```
mean(subset_data$number_of_obs);sd(subset_data$number_of_obs);min(subset_data$number_of_obs);max(subset_
```

```
## [1] 170.2517
```

```
## [1] 93.33147
```

```
## [1] 23
```

```
## [1] 261
```

Remove the legislation that deals with International Affairs

```
data111<-subset(data11,data11$policy.area!="International Affairs")
```

Models in the main text

```
packageVersion("rstanarm")
```

```
## [1] '2.21.4'
```

Model (1) in Table 3

```
data11$ia<-ifelse(data11$egan_owner_party2=="International Affairs",1,0)

model1<-stan_glmer(formula = complete_partisan~
                     #bill-level variables
                     ia+
                     abs(data11$dw_nominate_dim1)+
                     Legislation.type+
                     log(pages)+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level variables
                     number_of_obs+
                     (1|Congress),
                     data=data11,seed=921,
                     family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.305 seconds (Warm-up)
## Chain 1:                5.049 seconds (Sampling)
## Chain 1:                11.354 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 6.889 seconds (Warm-up)
## Chain 2:                6.826 seconds (Sampling)
## Chain 2:                13.715 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 12.506 seconds (Warm-up)
## Chain 3:                5.123 seconds (Sampling)
```

```
## Chain 3:                17.629 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 14.067 seconds (Warm-up)
## Chain 4:                10.53 seconds (Sampling)
## Chain 4:                24.597 seconds (Total)
## Chain 4:
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ia + abs(data11$dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 600
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                      -2.343  0.564 -3.264 -1.411
## ia                                               -0.755  0.209 -1.091 -0.404
```

```
## abs(data11$dw_nominate_dim1)                            3.755  0.603  2.798  4.736
## Legislation.typeResolution                              0.031  0.261 -0.405  0.466
## log(pages)                                             -0.121  0.114 -0.305  0.068
## term_has_served                                        -0.016  0.011 -0.034  0.002
## month_since_beginning_congress                          0.028  0.015  0.004  0.053
## number_of_obs                                           0.007  0.002  0.005  0.010
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.106  0.240 -0.577  0.150
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.059  0.204 -0.205  0.432
## b[(Intercept) Congress:113th_Congress_(2013-2014)] -0.009  0.213 -0.347  0.306
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.004  0.202 -0.318  0.315
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.042  0.194 -0.233  0.384
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.018  0.167 -0.225  0.304
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.005  0.233 -0.331  0.357
## Sigma[Congress:(Intercept),(Intercept)]             0.070  0.146  0.000  0.316
##
## Fit Diagnostics:
##           mean   sd    5%    95%
## mean_PPD 0.539  0.025 0.497 0.580
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                        mcse  Rhat  n_eff
## (Intercept)                                            0.010 1.001 2998
## ia                                                     0.003 1.000 4679
## abs(data11$dw_nominate_dim1)                           0.009 1.000 4085
## Legislation.typeResolution                             0.004 0.999 3976
## log(pages)                                             0.002 1.000 4057
## term_has_served                                        0.000 1.000 4641
## month_since_beginning_congress                         0.000 1.000 4065
## number_of_obs                                          0.000 1.006  740
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.005 1.001 1919
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.004 1.003 2179
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.004 1.000 2486
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.004 1.002 2513
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.004 1.001 2231
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.006 1.005  851
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.011 1.010  474
## Sigma[Congress:(Intercept),(Intercept)]            0.005 1.006 1035
## mean_PPD                                           0.000 1.001 4488
## log-posterior                                      0.087 1.001 1382
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
##Write the simulated coefficients output to an Excel file (change the path if necessary)##

#model_1<-as.data.frame(model1)
#write_xlsx(model_1,"C://isq//outside.xlsx")

##then save the file as a csv file##
```

Model (2) in Table 3

```
model2<-stan_glmer(formula = complete_partisan~
                     #bill-level variables
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+
                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level variables
                     number_of_obs+
                     (1|Congress),
                     data=data111,seed=921,
                     family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 8.008 seconds (Warm-up)
## Chain 1:                5.687 seconds (Sampling)
## Chain 1:                13.695 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 6.518 seconds (Warm-up)
## Chain 2:                3.302 seconds (Sampling)
## Chain 2:                9.82 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: Rejecting initial value:
## Chain 3:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 3:   Stan can't start sampling from this initial value.
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 7.433 seconds (Warm-up)
## Chain 3:                4.855 seconds (Sampling)
## Chain 3:                12.288 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 6.66 seconds (Warm-up)
## Chain 4:                6.07 seconds (Sampling)
## Chain 4:                12.73 seconds (Total)
## Chain 4:
```

```
## Warning: There were 5 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 256
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.713  0.927 -4.184 -1.215
## ownership_r_and_rowned                             0.664  0.368  0.041  1.269
## abs(dw_nominate_dim1)                              3.813  1.059  2.098  5.579
## log(pages)                                        -0.392  0.170 -0.678 -0.113
## Legislation.typeResolution                         0.900  0.745 -0.260  2.170
## term_has_served                                   -0.006  0.018 -0.036  0.024
## month_since_beginning_congress                     0.032  0.023 -0.006  0.070
## number_of_obs                                      0.008  0.003  0.004  0.013
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.301  0.504 -1.313  0.196
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.072  0.371 -0.447  0.728
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.135  0.426 -0.380  0.974
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.022  0.402 -0.688  0.604
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.099  0.350 -0.404  0.756
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.085  0.282 -0.328  0.610
## b[(Intercept) Congress:117th_Congress_(2021-2022)] -0.024  0.367 -0.655  0.582
## Sigma[Congress:(Intercept),(Intercept)]             0.233  0.442  0.000  0.959
```

```
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.676  0.037 0.613 0.734
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.014 1.000 4463
## ownership_r_and_rowned                             0.005 0.999 4578
## abs(dw_nominate_dim1)                              0.015 0.999 5039
## log(pages)                                         0.002 1.000 5221
## Legislation.typeResolution                         0.011 1.000 4543
## term_has_served                                    0.000 1.000 4493
## month_since_beginning_congress                     0.000 0.999 4746
## number_of_obs                                      0.000 1.000 2269
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.010 1.001 2369
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.006 0.999 3337
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.008 1.001 2545
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.006 1.000 4078
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.006 1.000 3157
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.006 0.999 2012
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.010 1.000 1382
## Sigma[Congress:(Intercept),(Intercept)]            0.010 1.002 1972
## mean_PPD                                           0.001 0.999 4671
## log-posterior                                      0.087 1.001 1382
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
##Write the simulated coefficients output to an Excel file (change the path if necessary)##

#data222<-as.data.frame(model2)
#write_xlsx(data222,"C://isq//owned.xlsx")
##then save the file as a csv file##
```

Figure 1

```
table(data11$Congress)
```

```
##
## 111th Congress (2009-2010) 112th Congress (2011-2012)
##                         38                         36
## 113th Congress (2013-2014) 114th Congress (2015-2016)
##                         30                         23
## 115th Congress (2017-2018) 116th Congress (2019-2020)
##                         49                        166
## 117th Congress (2021-2022)
##                        261
```

```r
data_number_by_congress<-data.frame(Congress=c("111th (2009-2010)","112th (2011-2012)","113th (2013-2014
                                               "114th (2015-2016)","115th (2017-2018)","116th (2019-2020
                                               "117th (2021-2022)"),
                                    number=c(38,36,30,23,49,166,261))
data_number_by_congress$Congress<-as.factor(data_number_by_congress$Congress)

ggplot(data = data_number_by_congress, mapping = aes(
  x = Congress, y = number )) +
  geom_col(width = 0.8) +
  xlab(" ") + ylab(" ") +
  theme(axis.line = element_line(colour = "black"),
        panel.background = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 21, face = "bold"),
        axis.text = element_text(size = 18, colour = "black"),
        axis.title.x = element_blank(),
        axis.text.x = element_text(angle = 0, vjust = 0.5, hjust = 0.5, size = 14)) +
  theme(panel.grid.major.x = element_blank()) +
  theme(panel.grid.major.y = element_line(colour = "grey28")) +
  ggtitle("Figure 1. The Number of China-related Legislation by Congress (2009-2022)") +
  scale_y_continuous(breaks = seq(0, 270, 30), limits = c(0, 270), minor_breaks = seq(0, 270, 30)) +
  labs(caption = "Note: The y-axis represents the number of measures. Legislation includes both bills (
  theme(plot.caption = element_text(size = 16, margin = margin(t = 20))) # Adjust the margin value as n
```

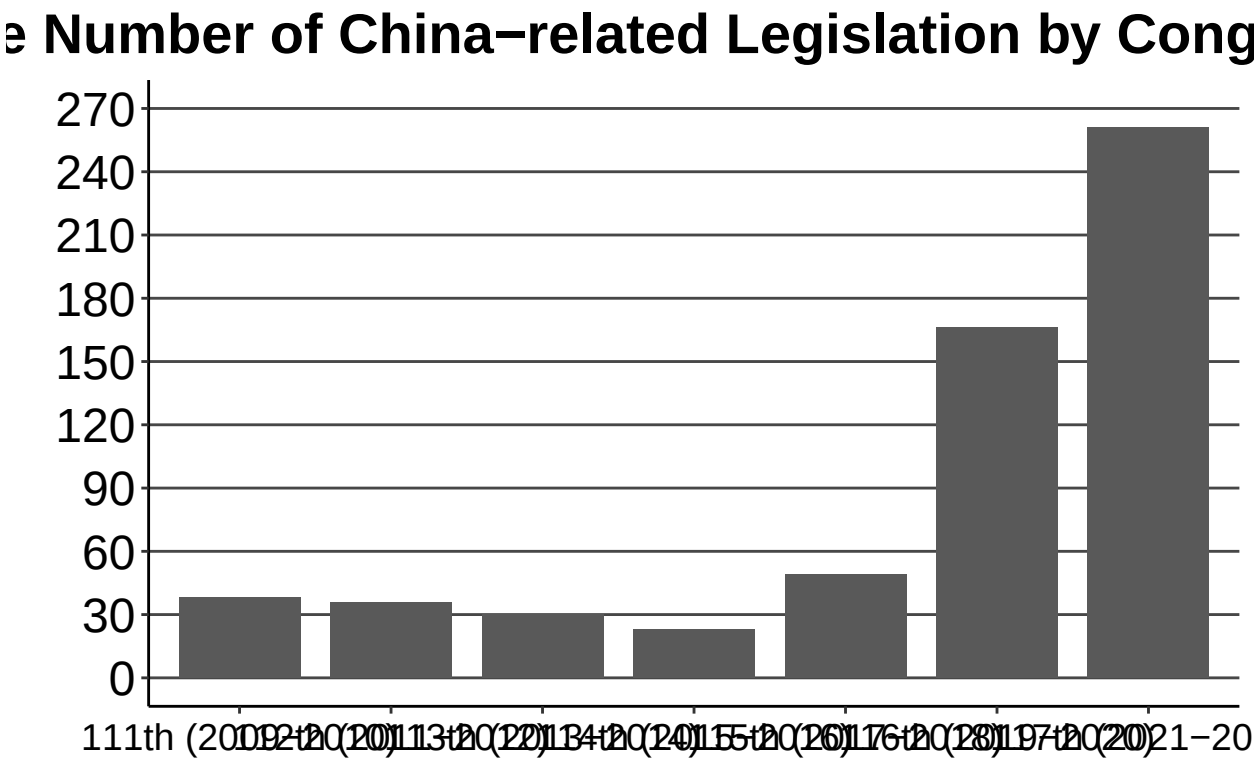


Figure 2

```
data_number_by_policyarea <- data.frame( Frequency = table(data111$policy.area))

data_number_by_policyarea <- data_number_by_policyarea[order(-data_number_by_policyarea$Frequency.Freq)

data_number_by_policyarea$Frequency.Var1<-factor(data_number_by_policyarea$Frequency.Var1,
                                                 levels = data_number_by_policyarea$Frequency.Var1[orde

ggplot(data = data_number_by_policyarea, mapping = aes(
  x=Frequency.Var1, y = Frequency.Freq))+
  geom_col(width = 0.8)+
  xlab(" ")+ylab(" ")+
  theme(axis.line = element_line(colour = "black"),
        panel.background = element_blank(),
        plot.title = element_text(hjust = 0.5,size=21,face = "bold"),
        axis.text=element_text(size=18,colour = "black"),
        axis.title.y=element_blank())+
  theme(panel.grid.major.y = element_blank())+
  coord_flip()+
  theme(panel.grid.major.x = element_line(colour="grey28"))+
  ggtitle("Figure 2. The Number of China-related Legislation by \nPolicy Areas (2009-2022)")+
  scale_y_continuous(breaks = seq(0,70,10),limits = c(0,70),minor_breaks = seq(0,70,10))+
  labs(caption = "Note: The x-axis represents the number of measures. Legislation includes both bills (
  theme(plot.caption = element_text(size = 16)) # Adjust the margin value as needed
```

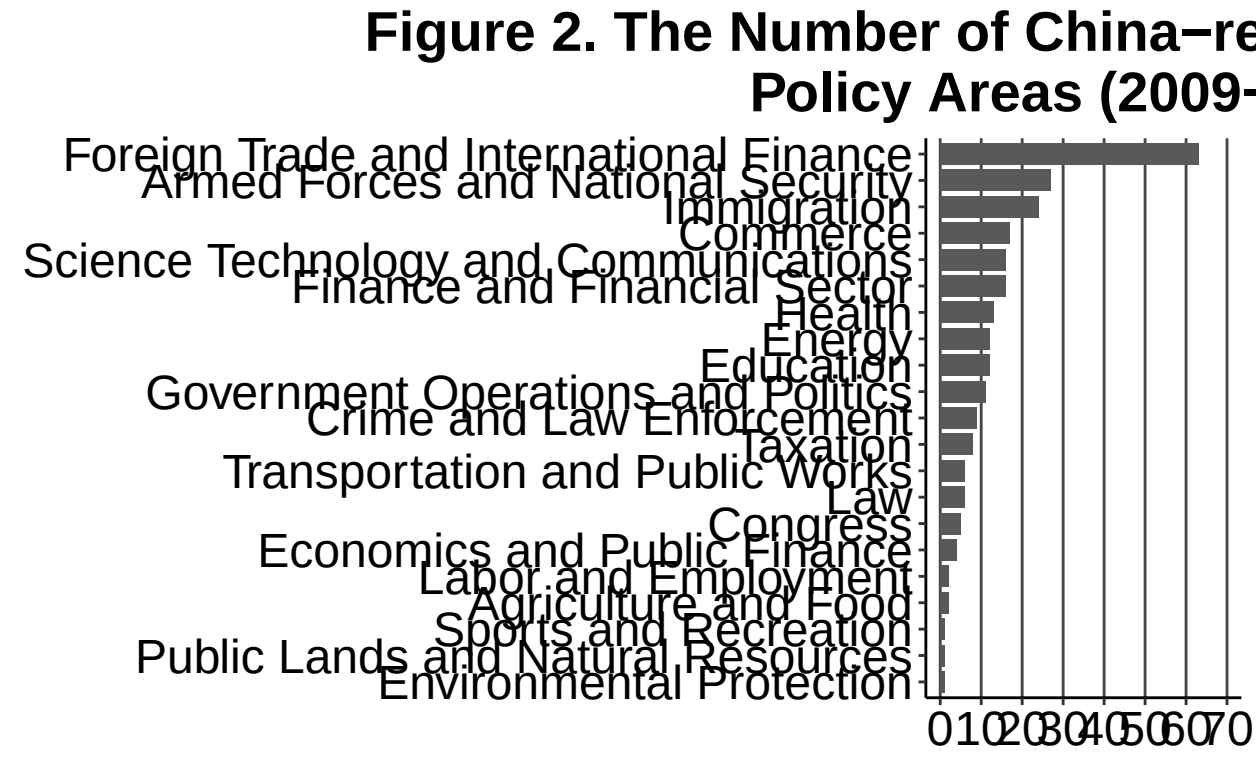


Figure 3

```
outside_coefficient<-read.csv("C://isq//outside.csv")
owned_coefficient<-read.csv("C://isq//owned.csv")

ia<-outside_coefficient$ia
owned<-owned_coefficient$ownership_r_and_rowned

layout(matrix(c(1, 2), 1, 2), widths=c(1, 1), heights=c(1))
hist(ia, xlab="Coefficient Value", main="3A. International Affairs",
     ylab="Frequency", breaks = 30,ylim = c(0,500))
abline(v=0, col="red", lwd=1.5, lty=2)
hist(owned, xlab="Coefficient Value", main="3B. Owned By One or Two Parties",
     ylab=" ", breaks = 30,ylim = c(0,500))
abline(v=0, col="red", lwd=1.5, lty=2)
mtext("Figure 3. Frequency Distribution of Simulated Coefficients Values",
      line = -1.5, cex=1.5, outer = TRUE,font = 2)
```

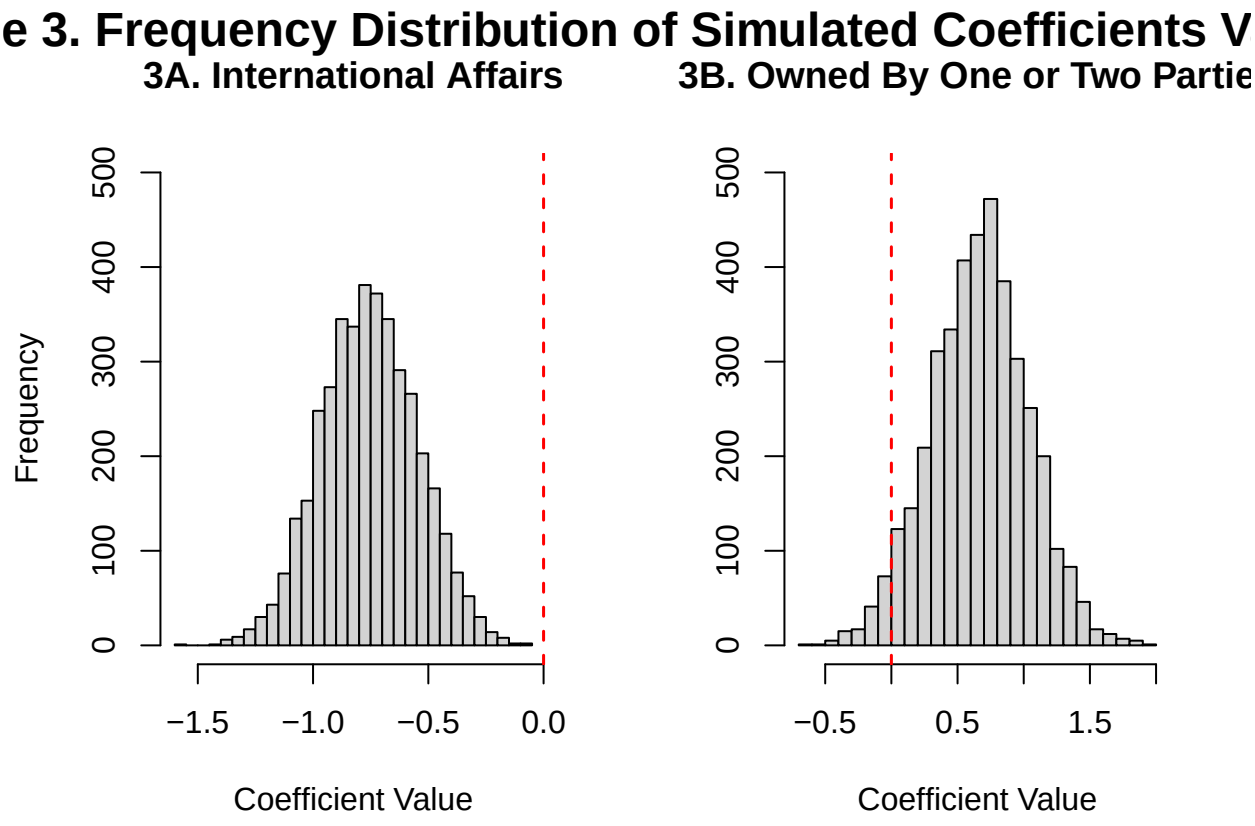


Figure 4

```
data_boxplot<-read.csv("C://isq//substantive significance.csv")
data_boxplot1<-data_boxplot[1:6,1:3]
```

```
plot1<-ggplot(data_boxplot1, aes(x=a, y=pred))+
  geom_point(position = position_dodge(width = 0.75),size=2.5,show.legend = FALSE,shape=15)+
  geom_line(position = position_dodge(width = 0.75))+theme_bw()+
  theme(axis.line = element_line(colour = "black"),
        panel.background = element_blank(),
        axis.text=element_text(size=13,colour = "black"),
        axis.title.x=element_blank(),
        plot.caption = element_text(size=16),
        legend.text = element_text(size=17),
        legend.title = element_text(size=17),
        plot.title = element_text(hjust = 0.5,size=15,face = "bold"),
        legend.spacing.x = unit(0.15,"cm"),
        legend.spacing.y = unit(0.1,"cm"),
        axis.title=element_text(size=15,face="bold"))+
  theme(legend.key.size = unit(1.3, 'cm'))+
  ylab(" ")+
  scale_y_continuous(breaks = seq(0.3,1,0.1),limits = c(0.3,1),minor_breaks = seq(0.3,1,.1))+
  theme(legend.position=c(0.15,0.16))+
  theme(panel.grid.major.x = element_blank())+
  theme(panel.grid.major.y = element_line(colour="grey28"))+
  scale_fill_brewer(palette = "Set3")+
  guides(size = guide_legend(nrow = 1))+
  #labs(x = "Bill length")+
  theme(plot.caption = element_text(hjust = 0)) +
  ggtitle("A. Domestic/International Affairs")+
  #ggtitle("A. Predicted Probabilities of Being Highly Partisan")+
  guides(shape = guide_legend(override.aes = list(size = 5)))+
  scale_x_discrete(labels = function(a) str_wrap(a, width = 15))

data_boxplot2<-data_boxplot[7:12,1:3]

plot2<-ggplot(data_boxplot2, aes(x=a, y=pred))+
  geom_point(position = position_dodge(width = 0.75),size=2.5,show.legend = FALSE,shape=15)+
  geom_line(position = position_dodge(width = 0.75))+theme_bw()+
  theme(axis.line = element_line(colour = "black"),
        panel.background = element_blank(),
        axis.text=element_text(size=13,colour = "black"),
        axis.title.x=element_blank(),
        plot.caption = element_text(size=16),
        legend.text = element_text(size=17),
        legend.title = element_text(size=17),
        plot.title = element_text(hjust = 0.5,size=15,face = "bold"),
        legend.spacing.x = unit(0.15,"cm"),
        legend.spacing.y = unit(0.1,"cm"),
        axis.title=element_text(size=15,face="bold"))+
  theme(legend.key.size = unit(1.3, 'cm'))+
  ylab(" ")+
  scale_y_continuous(breaks = seq(0.3,1,0.1),limits = c(0.3,1),minor_breaks = seq(0.3,1,.1))+
  theme(legend.position=c(0.15,0.16))+
  theme(panel.grid.major.x = element_blank())+
  theme(panel.grid.major.y = element_line(colour="grey28"))+
  scale_fill_brewer(palette = "Set3")+
  guides(size = guide_legend(nrow = 1))+
```

```
  #labs(x = "Bill length")+
  theme(plot.caption = element_text(hjust = 0)) +
  ggtitle("B. Issue Ownership")+
  #ggtitle("A. Predicted Probabilities of Being Highly Partisan")+
  guides(shape = guide_legend(override.aes = list(size = 5)))+
  scale_x_discrete(labels = function(a) str_wrap(a, width = 15))

library(ggpubr)
```

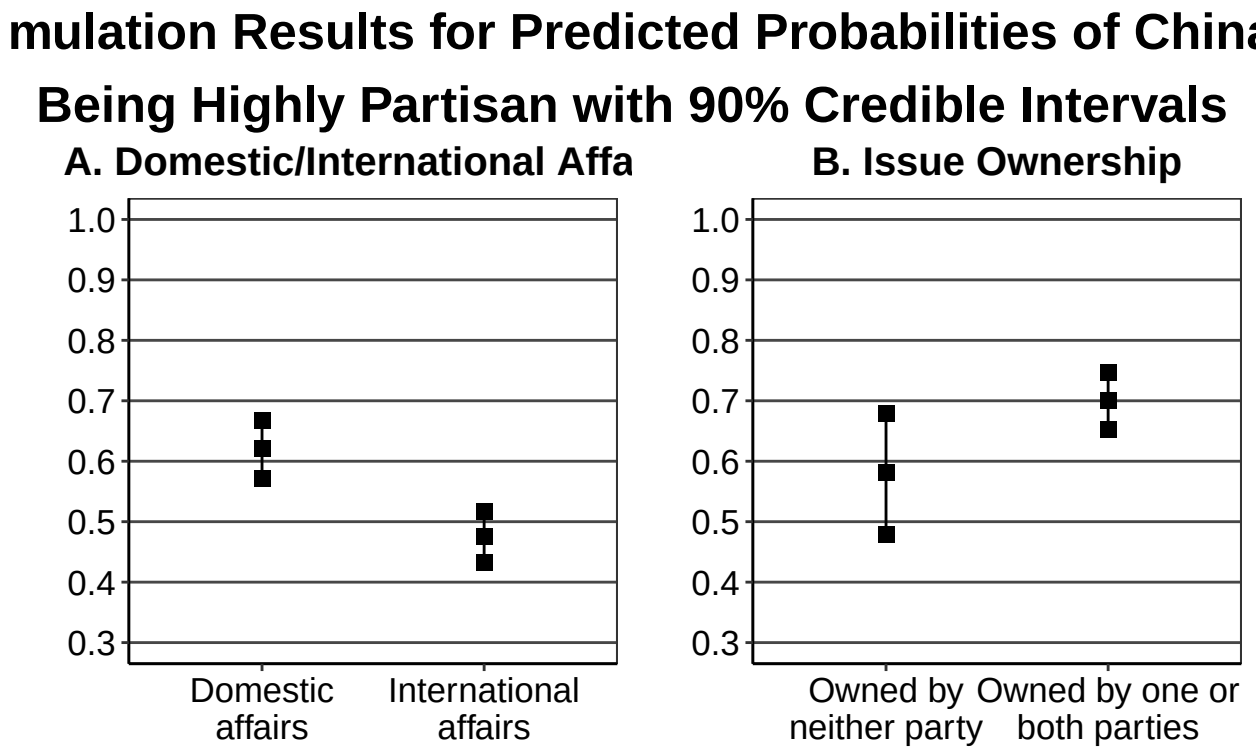
```
##
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:plyr':
##
##     mutate
```

```
library(stringr)

taiwan_figure<-ggarrange(plot1,plot2,
                         ncol = 2, nrow = 1)
annotate_figure(taiwan_figure,top = text_grob("Figure 4. Statistical Simulation Results for Predicted Pr
                bottom = text_grob("Note: Figure 4A and 4B are based on Model (1) and (2) in Table 3, re

                                   ,hjust=1,x=1,size = 15))
```



Appendix

Table A3

```
model1<-stan_glmer(formula = complete_partisan4~
                     ia+
                     abs(dw_nominate_dim1)+
                     Legislation.type+
                     log(pages)+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data11,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.002 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 14.877 seconds (Warm-up)
## Chain 1:                7.499 seconds (Sampling)
## Chain 1:                22.376 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 9.609 seconds (Warm-up)
## Chain 2:                7.722 seconds (Sampling)
## Chain 2:                17.331 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 14.494 seconds (Warm-up)
## Chain 3:                21.141 seconds (Sampling)
## Chain 3:                35.635 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 10.87 seconds (Warm-up)
## Chain 4:                11.889 seconds (Sampling)
## Chain 4:                22.759 seconds (Total)
## Chain 4:
```

```
## Warning: There were 3 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan4 ~ ia + abs(dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 600
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -0.098  0.664 -1.183  0.979
## ia                                                -1.159  0.285 -1.634 -0.703
## abs(dw_nominate_dim1)                              3.213  0.725  2.016  4.411
## Legislation.typeResolution                        -0.312  0.283 -0.766  0.162
## log(pages)                                        -0.066  0.134 -0.289  0.157
## term_has_served                                   -0.014  0.012 -0.034  0.006
## month_since_beginning_congress                     0.072  0.018  0.042  0.101
## number_of_obs                                      0.003  0.002 -0.001  0.006
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.214  0.317 -0.835  0.159
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.023  0.277 -0.425  0.488
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.119  0.299 -0.290  0.676
## b[(Intercept) Congress:114th_Congress_(2015-2016)]  0.112  0.318 -0.328  0.657
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.043  0.265 -0.367  0.496
## b[(Intercept) Congress:116th_Congress_(2019-2020)] -0.188  0.272 -0.679  0.153
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.126  0.362 -0.341  0.770
## Sigma[Congress:(Intercept),(Intercept)]             0.163  0.276  0.001  0.602
##
```

```
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.796  0.021 0.762 0.830
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.011 1.000 3447
## ia                                                 0.004 1.000 4214
## abs(dw_nominate_dim1)                              0.011 1.000 4510
## Legislation.typeResolution                         0.004 1.000 4065
## log(pages)                                         0.002 1.000 4490
## term_has_served                                    0.000 1.000 3545
## month_since_beginning_congress                     0.000 0.999 4276
## number_of_obs                                      0.000 1.000 1569
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.006 1.000 2781
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.005 1.000 3754
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.006 1.000 2888
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.006 1.001 2624
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.004 1.001 3512
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.007 0.999 1741
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.010 1.000 1337
## Sigma[Congress:(Intercept),(Intercept)]            0.009 1.004  998
## mean_PPD                                           0.000 1.000 4313
## log-posterior                                      0.095 1.004 1231
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan4~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data111,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.119 seconds (Warm-up)
## Chain 1:                2.142 seconds (Sampling)
## Chain 1:                8.261 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 2.259 seconds (Warm-up)
## Chain 2:                2.099 seconds (Sampling)
## Chain 2:                4.358 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 2.233 seconds (Warm-up)
## Chain 3:                1.781 seconds (Sampling)
## Chain 3:                4.014 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 2.516 seconds (Warm-up)
## Chain 4:                2.214 seconds (Sampling)
## Chain 4:                4.73 seconds (Total)
## Chain 4:
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan4 ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 256
##  groups:       Congress (7)
##
## Estimates:
```

```
##                                                    mean   sd     5%     95%
## (Intercept)                                      -1.494  1.336 -3.700  0.674
## ownership_r_and_rowned                            0.986  0.546  0.073  1.886
## abs(dw_nominate_dim1)                             4.359  1.671  1.677  7.126
## log(pages)                                       -0.198  0.255 -0.612  0.224
## Legislation.typeResolution                        0.454  1.291 -1.332  2.861
## term_has_served                                   0.073  0.035  0.020  0.133
## month_since_beginning_congress                    0.037  0.037 -0.022  0.099
## number_of_obs                                     0.004  0.004 -0.003  0.010
## b[(Intercept) Congress:111th_Congress_(2009-2010)]  0.024  0.471 -0.649  0.802
## b[(Intercept) Congress:112th_Congress_(2011-2012)] -0.084  0.456 -0.870  0.555
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.162  0.540 -0.456  1.132
## b[(Intercept) Congress:114th_Congress_(2015-2016)]  0.090  0.553 -0.645  0.991
## b[(Intercept) Congress:115th_Congress_(2017-2018)] -0.141  0.476 -1.022  0.436
## b[(Intercept) Congress:116th_Congress_(2019-2020)] -0.025  0.374 -0.647  0.583
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.136  0.542 -0.528  1.088
## Sigma[Congress:(Intercept),(Intercept)]           0.324  0.720  0.001  1.410
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.912  0.023 0.871 0.945
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                  mcse  Rhat  n_eff
## (Intercept)                                      0.020 1.000 4289
## ownership_r_and_rowned                           0.007 1.000 5540
## abs(dw_nominate_dim1)                            0.023 0.999 5169
## log(pages)                                       0.004 1.000 5261
## Legislation.typeResolution                       0.022 1.000 3321
## term_has_served                                  0.001 1.000 4423
## month_since_beginning_congress                   0.001 0.999 5045
## number_of_obs                                    0.000 1.002 2131
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.007 1.000 4556
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.008 1.001 3596
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.009 1.000 3366
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.009 0.999 3752
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.009 1.000 2974
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.009 1.002 1774
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.016 1.006 1194
## Sigma[Congress:(Intercept),(Intercept)]          0.017 1.002 1797
## mean_PPD                                         0.000 1.001 4172
## log-posterior                                    0.080 1.002 1462
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Table A4

```
model1<-stan_glmer(formula = complete_partisan3~
                     ia+
                     abs(dw_nominate_dim1)+
```

```
                    Legislation.type+
                    log(pages)+
                    term_has_served+
                    month_since_beginning_congress+

                    #group-level
                    number_of_obs+
                    (1|Congress),
                  data=data11,seed=921,
                  family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.003 seconds (Warm-up)
## Chain 1:                3.7 seconds (Sampling)
## Chain 1:                9.703 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 5.599 seconds (Warm-up)
## Chain 2:                6.403 seconds (Sampling)
## Chain 2:                12.002 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 6.144 seconds (Warm-up)
## Chain 3:                3.989 seconds (Sampling)
## Chain 3:                10.133 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
```

```
## Chain 4:  Elapsed Time: 9.34 seconds (Warm-up)
## Chain 4:                11.783 seconds (Sampling)
## Chain 4:                21.123 seconds (Total)
## Chain 4:
```

```
## Warning: There were 3 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan3 ~ ia + abs(dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 600
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                      -1.184  0.652 -2.251 -0.114
## ia                                               -0.971  0.248 -1.371 -0.571
## abs(dw_nominate_dim1)                             3.756  0.719  2.563  4.938
## Legislation.typeResolution                       -0.374  0.267 -0.820  0.081
## log(pages)                                       -0.013  0.126 -0.216  0.196
## term_has_served                                  -0.007  0.012 -0.027  0.012
## month_since_beginning_congress                    0.066  0.017  0.038  0.094
## number_of_obs                                     0.005  0.002  0.001  0.008
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.106  0.259 -0.567  0.224
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.011  0.249 -0.385  0.420
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.075  0.260 -0.291  0.539
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.063  0.270 -0.550  0.337
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.162  0.273 -0.162  0.676
## b[(Intercept) Congress:116th_Congress_(2019-2020)] -0.123  0.230 -0.531  0.176
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.088  0.312 -0.343  0.610
## Sigma[Congress:(Intercept),(Intercept)]             0.124  0.242  0.001  0.480
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.748  0.022 0.712 0.783
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                 mcse  Rhat  n_eff
```

```
## (Intercept)                                          0.012 1.000 2766
## ia                                                   0.004 1.000 3365
## abs(dw_nominate_dim1)                                0.012 1.000 3760
## Legislation.typeResolution                           0.005 1.000 3326
## log(pages)                                           0.002 1.000 3964
## term_has_served                                      0.000 1.000 3860
## month_since_beginning_congress                       0.000 1.000 3695
## number_of_obs                                        0.000 1.003  788
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.006 1.003 1994
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.006 1.002 1792
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.006 1.002 1933
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.006 1.001 1854
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.007 1.002 1498
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.007 1.003 1242
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.012 1.004  700
## Sigma[Congress:(Intercept),(Intercept)]              0.010 1.011  545
## mean_PPD                                             0.000 1.000 4382
## log-posterior                                        0.093 1.006 1268
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan3~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data111,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.893 seconds (Warm-up)
## Chain 1:                4.531 seconds (Sampling)
## Chain 1:                11.424 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 2.356 seconds (Warm-up)
## Chain 2:                1.598 seconds (Sampling)
## Chain 2:                3.954 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
```

```
## Chain 3:  Elapsed Time: 2.222 seconds (Warm-up)
## Chain 3:                2.278 seconds (Sampling)
## Chain 3:                4.5 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 2.401 seconds (Warm-up)
## Chain 4:                2.218 seconds (Sampling)
## Chain 4:                4.619 seconds (Total)
## Chain 4:
```

```
## Warning: There were 3 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan3 ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 256
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
```

```
## (Intercept)                                          -2.228  1.182 -4.172 -0.279
## ownership_r_and_rowned                                0.512  0.466 -0.247  1.254
## abs(dw_nominate_dim1)                                 4.258  1.390  2.040  6.580
## log(pages)                                           -0.025  0.230 -0.392  0.358
## Legislation.typeResolution                           -0.123  0.900 -1.462  1.492
## term_has_served                                       0.056  0.028  0.011  0.104
## month_since_beginning_congress                        0.037  0.031 -0.014  0.088
## number_of_obs                                         0.006  0.004  0.000  0.012
## b[(Intercept) Congress:111th_Congress_(2009-2010)]    0.002  0.435 -0.679  0.658
## b[(Intercept) Congress:112th_Congress_(2011-2012)]    0.018  0.429 -0.610  0.668
## b[(Intercept) Congress:113th_Congress_(2013-2014)]    0.239  0.571 -0.308  1.324
## b[(Intercept) Congress:114th_Congress_(2015-2016)]   -0.219  0.541 -1.319  0.353
## b[(Intercept) Congress:115th_Congress_(2017-2018)]    0.064  0.423 -0.537  0.792
## b[(Intercept) Congress:116th_Congress_(2019-2020)]   -0.048  0.362 -0.629  0.489
## b[(Intercept) Congress:117th_Congress_(2021-2022)]    0.089  0.513 -0.531  0.971
## Sigma[Congress:(Intercept),(Intercept)]               0.303  0.671  0.001  1.373
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.873  0.028 0.824 0.918
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                      mcse  Rhat  n_eff
## (Intercept)                                          0.022 1.000 2802
## ownership_r_and_rowned                               0.007 0.999 4060
## abs(dw_nominate_dim1)                                0.023 0.999 3805
## log(pages)                                           0.003 1.000 4824
## Legislation.typeResolution                           0.016 1.000 3314
## term_has_served                                      0.000 1.000 3659
## month_since_beginning_congress                       0.000 1.000 4043
## number_of_obs                                        0.000 1.006  986
## b[(Intercept) Congress:111th_Congress_(2009-2010)]   0.012 1.003 1369
## b[(Intercept) Congress:112th_Congress_(2011-2012)]   0.013 1.004 1047
## b[(Intercept) Congress:113th_Congress_(2013-2014)]   0.017 1.004 1066
## b[(Intercept) Congress:114th_Congress_(2015-2016)]   0.013 1.000 1743
## b[(Intercept) Congress:115th_Congress_(2017-2018)]   0.010 1.002 1658
## b[(Intercept) Congress:116th_Congress_(2019-2020)]   0.009 1.002 1478
## b[(Intercept) Congress:117th_Congress_(2021-2022)]   0.017 1.005  877
## Sigma[Congress:(Intercept),(Intercept)]              0.022 1.003  915
## mean_PPD                                             0.000 1.000 4434
## log-posterior                                        0.076 1.000 1679
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Table A5

```
model1<-stan_glmer(formula = complete_partisan2~
                     ia+
                     abs(dw_nominate_dim1)+
                     Legislation.type+
```

```
                    log(pages)+
                    term_has_served+
                    month_since_beginning_congress+

                    #group-level
                    number_of_obs+
                    (1|Congress),
                  data=data11,seed=921,
                  family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.136 seconds (Warm-up)
## Chain 1:                3.698 seconds (Sampling)
## Chain 1:                9.834 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 5.674 seconds (Warm-up)
## Chain 2:                5.081 seconds (Sampling)
## Chain 2:                10.755 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 6.264 seconds (Warm-up)
## Chain 3:                5.233 seconds (Sampling)
## Chain 3:                11.497 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 11.347 seconds (Warm-up)
```

```
## Chain 4:                10.529 seconds (Sampling)
## Chain 4:                21.876 seconds (Total)
## Chain 4:
```

```
## Warning: There were 2 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan2 ~ ia + abs(dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 600
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                      -1.589  0.601 -2.581 -0.616
## ia                                               -0.867  0.241 -1.263 -0.471
## abs(dw_nominate_dim1)                             3.419  0.658  2.323  4.495
## Legislation.typeResolution                       -0.220  0.266 -0.657  0.213
## log(pages)                                        0.042  0.121 -0.156  0.245
## term_has_served                                  -0.009  0.011 -0.026  0.008
## month_since_beginning_congress                    0.056  0.016  0.030  0.082
## number_of_obs                                     0.006  0.002  0.002  0.009
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.101  0.267 -0.587  0.292
## b[(Intercept) Congress:112th_Congress_(2011-2012)] -0.039  0.265 -0.495  0.376
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.161  0.295 -0.234  0.703
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.069  0.282 -0.570  0.352
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.138  0.256 -0.209  0.606
## b[(Intercept) Congress:116th_Congress_(2019-2020)] -0.198  0.247 -0.658  0.131
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.133  0.332 -0.327  0.719
## Sigma[Congress:(Intercept),(Intercept)]            0.146  0.246  0.001  0.508
##
## Fit Diagnostics:
##           mean   sd    5%    95%
## mean_PPD 0.700  0.023 0.660 0.738
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                  mcse  Rhat  n_eff
## (Intercept)                                      0.010 1.001 3369
```

```
## ia                                                    0.004 0.999 4080
## abs(dw_nominate_dim1)                                 0.010 1.001 4080
## Legislation.typeResolution                            0.004 1.000 3872
## log(pages)                                            0.002 1.001 3883
## term_has_served                                       0.000 1.000 3845
## month_since_beginning_congress                        0.000 1.000 3714
## number_of_obs                                         0.000 1.002 1440
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.005 1.000 3026
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.005 1.000 2435
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.006 1.000 2612
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.005 1.000 3616
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.005 1.001 2963
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.006 1.000 1476
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.010 1.003 1152
## Sigma[Congress:(Intercept),(Intercept)]            0.006 1.004 1455
## mean_PPD                                              0.000 1.000 4009
## log-posterior                                         0.100 1.002 1127
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan2~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data111,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.116 seconds (Warm-up)
## Chain 1:                4.633 seconds (Sampling)
## Chain 1:                10.749 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 5.802 seconds (Warm-up)
## Chain 2:                4.724 seconds (Sampling)
## Chain 2:                10.526 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 6.109 seconds (Warm-up)
```

```
## Chain 3:                4.512 seconds (Sampling)
## Chain 3:                10.621 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 6.617 seconds (Warm-up)
## Chain 4:                5.585 seconds (Sampling)
## Chain 4:                12.202 seconds (Total)
## Chain 4:
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan2 ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 256
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean    sd     5%     95%
## (Intercept)                                      -2.244  1.017 -3.926 -0.583
```

```
## ownership_r_and_rowned                                  0.893  0.417  0.204  1.580
## abs(dw_nominate_dim1)                                   3.480  1.229  1.443  5.487
## log(pages)                                             -0.151  0.192 -0.454  0.167
## Legislation.typeResolution                              0.455  0.900 -0.892  2.063
## term_has_served                                         0.024  0.021 -0.010  0.060
## month_since_beginning_congress                          0.032  0.028 -0.013  0.079
## number_of_obs                                           0.007  0.003  0.002  0.012
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.020  0.358 -0.630  0.548
## b[(Intercept) Congress:112th_Congress_(2011-2012)] -0.038  0.353 -0.637  0.504
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.134  0.411 -0.388  0.916
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.097  0.405 -0.835  0.447
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.076  0.349 -0.418  0.720
## b[(Intercept) Congress:116th_Congress_(2019-2020)] -0.157  0.317 -0.743  0.235
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.124  0.440 -0.431  0.882
## Sigma[Congress:(Intercept),(Intercept)]                 0.223  0.456  0.000  0.881
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.826  0.030 0.773 0.875
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.017 1.001 3534
## ownership_r_and_rowned                             0.006 1.000 4461
## abs(dw_nominate_dim1)                              0.017 1.000 5230
## log(pages)                                         0.003 1.000 5131
## Legislation.typeResolution                         0.014 1.000 3907
## term_has_served                                    0.000 1.001 4344
## month_since_beginning_congress                     0.000 0.999 4982
## number_of_obs                                      0.000 1.003 1725
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.006 0.999 4195
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.006 1.000 3537
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.007 1.000 3489
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.007 1.001 3169
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.006 1.000 3829
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.008 1.005 1448
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.013 1.005 1229
## Sigma[Congress:(Intercept),(Intercept)]            0.013 1.002 1279
## mean_PPD                                           0.000 1.001 4149
## log-posterior                                      0.079 1.000 1578
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Table A6

```
data1<-read.csv("C://isq//china bills 2009-2022.csv",fileEncoding="UTF-8-BOM")

data1 <- subset(data1, !policy.area %in% c("Foreign Trade and International Finance",
                                           "Armed Service and National Security"))
```

```
data1$dw_nominate_dim1<-as.numeric(data1$dw_nominate_dim1)
```

```
## Warning: NAs introduced by coercion
```

```
data1$pages<-as.numeric(data1$pages)
```

```
## Warning: NAs introduced by coercion
```

```
data1$Legislation.type<-ifelse(data1$Legislation.type=="Bill","Bill","Resolution")
data1$term_has_served<-as.numeric(data1$term_has_served)
data1$month_since_beginning_congress<-as.numeric(data1$month_since_beginning_congress)

data11<-subset(data1,data1$anti.china==1)
nrow(data11)  #540
```

```
## [1] 540
```

```
table(data11$Congress)
```

```
##
## 111th Congress (2009-2010) 112th Congress (2011-2012)
##                         32                         32
## 113th Congress (2013-2014) 114th Congress (2015-2016)
##                         29                         21
## 115th Congress (2017-2018) 116th Congress (2019-2020)
##                         42                        148
## 117th Congress (2021-2022)
##                        236
```

```
number_of_obs <- ifelse(data11$Congress == "111th Congress (2009-2010)", 38,
                        ifelse(data11$Congress == "112th Congress (2011-2012)", 36,
                               ifelse(data11$Congress == "113th Congress (2013-2014)", 30,
                                      ifelse(data11$Congress == "114th Congress (2015-2016)", 23,
                                             ifelse(data11$Congress == "115th Congress (2017-2018)", 49
                                                    ifelse(data11$Congress == "116th Congress (2019-202
                                                           ifelse(data11$Congress == "117th Congress (2

data11$number_of_obs <- as.numeric(number_of_obs)

data111<-subset(data11,data11$policy.area!="International Affairs")
nrow(data111)   #193
```

```
## [1] 193
```

```
data11$ia<-ifelse(data11$egan_owner_party2=="International Affairs",1,0)

model1<-stan_glmer(formula = complete_partisan~
                     ia+
                     abs(data11$dw_nominate_dim1)+
```

```
                    Legislation.type+
                    log(pages)+
                    term_has_served+
                    month_since_beginning_congress+

                    #group-level
                    number_of_obs+
                    (1|Congress),
                    data=data11,seed=921,
                    family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 12.164 seconds (Warm-up)
## Chain 1:                9.211 seconds (Sampling)
## Chain 1:                21.375 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 13.674 seconds (Warm-up)
## Chain 2:                8.557 seconds (Sampling)
## Chain 2:                22.231 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 14.909 seconds (Warm-up)
## Chain 3:                11.003 seconds (Sampling)
## Chain 3:                25.912 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
```

```
## Chain 4:  Elapsed Time: 13.592 seconds (Warm-up)
## Chain 4:                6.385 seconds (Sampling)
## Chain 4:                19.977 seconds (Total)
## Chain 4:
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ia + abs(data11$dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 537
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.136  0.577 -3.078 -1.199
## ia                                                -0.772  0.230 -1.155 -0.400
## abs(data11$dw_nominate_dim1)                       3.657  0.643  2.599  4.753
## Legislation.typeResolution                        -0.019  0.273 -0.465  0.432
## log(pages)                                        -0.062  0.117 -0.254  0.135
## term_has_served                                   -0.022  0.011 -0.041 -0.004
## month_since_beginning_congress                     0.021  0.015 -0.004  0.045
## number_of_obs                                      0.007  0.002  0.004  0.009
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.058  0.216 -0.452  0.215
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.045  0.195 -0.221  0.393
## b[(Intercept) Congress:113th_Congress_(2013-2014)] -0.030  0.194 -0.351  0.236
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.011  0.206 -0.329  0.288
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.049  0.188 -0.198  0.396
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.006  0.150 -0.223  0.262
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.008  0.195 -0.287  0.317
## Sigma[Congress:(Intercept),(Intercept)]             0.059  0.137  0.000  0.245
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.527  0.026 0.484 0.572
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                mcse  Rhat  n_eff
```

```
## (Intercept)                                          0.009 1.001 3742
## ia                                                   0.004 0.999 4245
## abs(data11$dw_nominate_dim1)                         0.010 1.000 4560
## Legislation.typeResolution                           0.004 1.000 3866
## log(pages)                                           0.002 1.000 4074
## term_has_served                                      0.000 1.000 4468
## month_since_beginning_congress                       0.000 1.000 4881
## number_of_obs                                        0.000 1.002 2296
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.004 1.001 2399
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.004 1.001 2999
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.003 1.000 3318
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.004 1.001 3229
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.004 1.001 2831
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.003 1.003 2118
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.005 1.004 1486
## Sigma[Congress:(Intercept),(Intercept)]              0.003 1.000 1554
## mean_PPD                                             0.000 0.999 4159
## log-posterior                                        0.079 1.000 1545
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                     data=data111,seed=921,
                     family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 5.02 seconds (Warm-up)
## Chain 1:                4.359 seconds (Sampling)
## Chain 1:                9.379 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 4.604 seconds (Warm-up)
## Chain 2:                3.622 seconds (Sampling)
## Chain 2:                8.226 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: Rejecting initial value:
## Chain 3:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 3:   Stan can't start sampling from this initial value.
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 4.613 seconds (Warm-up)
## Chain 3:                2.933 seconds (Sampling)
## Chain 3:                7.546 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 5.578 seconds (Warm-up)
## Chain 4:                5.395 seconds (Sampling)
## Chain 4:                10.973 seconds (Total)
## Chain 4:
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 193
##  groups:       Congress (7)
```

```
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.016  1.021 -3.737 -0.364
## ownership_r_and_rowned                             0.700  0.387  0.051  1.336
## abs(dw_nominate_dim1)                              3.675  1.201  1.702  5.649
## log(pages)                                        -0.285  0.188 -0.609  0.018
## Legislation.typeResolution                         1.890  1.402 -0.096  4.443
## term_has_served                                   -0.032  0.022 -0.069  0.003
## month_since_beginning_congress                     0.012  0.028 -0.035  0.056
## number_of_obs                                      0.006  0.003  0.001  0.012
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.186  0.535 -1.253  0.389
## b[(Intercept) Congress:112th_Congress_(2011-2012)] -0.002  0.378 -0.612  0.587
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.004  0.414 -0.640  0.671
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.055  0.424 -0.779  0.510
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.207  0.468 -0.275  1.141
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.030  0.303 -0.441  0.565
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.005  0.406 -0.611  0.664
## Sigma[Congress:(Intercept),(Intercept)]             0.260  0.572  0.000  1.171
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.689  0.041 0.622 0.756
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.016 1.000 4091
## ownership_r_and_rowned                             0.006 1.000 4851
## abs(dw_nominate_dim1)                              0.019 1.000 3939
## log(pages)                                         0.003 0.999 4252
## Legislation.typeResolution                         0.024 1.000 3549
## term_has_served                                    0.000 0.999 3908
## month_since_beginning_congress                     0.000 1.000 4245
## number_of_obs                                      0.000 1.001 2185
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.011 1.001 2537
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.006 1.000 3833
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.007 1.000 3634
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.007 0.999 3696
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.009 1.000 2611
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.007 1.002 1856
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.012 1.003 1179
## Sigma[Congress:(Intercept),(Intercept)]            0.014 1.000 1617
## mean_PPD                                           0.001 1.000 3683
## log-posterior                                      0.086 1.002 1378
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Table A7

```
data1<-read.csv("C://isq//china bills 2009-2022.csv",fileEncoding="UTF-8-BOM")

data1$dw_nominate_dim1<-as.numeric(data1$dw_nominate_dim1)
```

```
## Warning: NAs introduced by coercion
```

```
data1$pages<-as.numeric(data1$pages)
```

```
## Warning: NAs introduced by coercion
```

```
data1$Legislation.type<-ifelse(data1$Legislation.type=="Bill","Bill","Resolution")
data1$term_has_served<-as.numeric(data1$term_has_served)
data1$month_since_beginning_congress<-as.numeric(data1$month_since_beginning_congress)

data11<-subset(data1,data1$anti.china==1)
nrow(data11)  #603
```

```
## [1] 603
```

```
table(data11$Congress)
```

```
##
## 111th Congress (2009-2010) 112th Congress (2011-2012)
##                         38                         36
## 113th Congress (2013-2014) 114th Congress (2015-2016)
##                         30                         23
## 115th Congress (2017-2018) 116th Congress (2019-2020)
##                         49                        166
## 117th Congress (2021-2022)
##                        261
```

```
number_of_obs <- ifelse(data11$Congress == "111th Congress (2009-2010)", 38,
                        ifelse(data11$Congress == "112th Congress (2011-2012)", 36,
                               ifelse(data11$Congress == "113th Congress (2013-2014)", 30,
                                      ifelse(data11$Congress == "114th Congress (2015-2016)", 23,
                                             ifelse(data11$Congress == "115th Congress (2017-2018)", 49
                                                    ifelse(data11$Congress == "116th Congress (2019-202
                                                           ifelse(data11$Congress == "117th Congress (2

data11$number_of_obs <- as.numeric(number_of_obs)

data111 <- subset(data11, !policy.area %in% c("Foreign Trade and International Finance",
                                              "Armed Service and National Security",
                                              "International Affairs"))
nrow(data111)   #193
```

```
## [1] 193
```

```r
data11$ia <- ifelse(data11$policy.area %in% c("International Affairs",
                                              "Foreign Trade and International Finance",
                                              "Armed Forces and National Security"), 1, 0)

table(data11$ia)
```

```
##
##   0   1
## 166 437
```

```r
model1<-stan_glmer(formula = complete_partisan~
                     ia+
                     abs(data11$dw_nominate_dim1)+
                     Legislation.type+
                     log(pages)+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data11,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 8.834 seconds (Warm-up)
## Chain 1:                11.9 seconds (Sampling)
## Chain 1:                20.734 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
```

```
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 15.78 seconds (Warm-up)
## Chain 2:                9.607 seconds (Sampling)
## Chain 2:                25.387 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 15.096 seconds (Warm-up)
## Chain 3:                9.909 seconds (Sampling)
## Chain 3:                25.005 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
```

```
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 15.811 seconds (Warm-up)
## Chain 4:                9.195 seconds (Sampling)
## Chain 4:                25.006 seconds (Total)
## Chain 4:
```

```
## Warning: There were 3 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ia + abs(data11$dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 600
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.188  0.556 -3.092 -1.282
## ia                                                -0.681  0.231 -1.063 -0.302
## abs(data11$dw_nominate_dim1)                       3.849  0.651  2.797  4.935
## Legislation.typeResolution                        -0.089  0.256 -0.502  0.337
## log(pages)                                        -0.120  0.113 -0.305  0.063
## term_has_served                                   -0.018  0.011 -0.036 -0.001
## month_since_beginning_congress                     0.027  0.014  0.004  0.051
## number_of_obs                                      0.007  0.002  0.004  0.009
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.097  0.230 -0.518  0.157
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.065  0.209 -0.201  0.457
## b[(Intercept) Congress:113th_Congress_(2013-2014)] -0.013  0.191 -0.325  0.291
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.014  0.214 -0.357  0.310
```

```
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.052  0.186 -0.201  0.384
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.012  0.170 -0.236  0.292
## b[(Intercept) Congress:117th_Congress_(2021-2022)] -0.005  0.243 -0.360  0.340
## Sigma[Congress:(Intercept),(Intercept)]             0.067  0.141  0.000  0.290
##
## Fit Diagnostics:
##           mean   sd    5%    95%
## mean_PPD 0.539  0.025 0.497 0.578
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                     mcse  Rhat  n_eff
## (Intercept)                                         0.009 1.000 3893
## ia                                                  0.004 1.000 4224
## abs(data11$dw_nominate_dim1)                        0.010 1.000 4161
## Legislation.typeResolution                          0.004 1.001 3665
## log(pages)                                          0.002 1.000 3945
## term_has_served                                     0.000 1.000 4217
## month_since_beginning_congress                      0.000 1.001 4383
## number_of_obs                                       0.000 1.003  870
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.006 1.001 1726
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.004 0.999 3068
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.004 1.001 2930
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.004 1.000 2889
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.003 1.000 3028
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.006 1.008  766
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.011 1.009  500
## Sigma[Congress:(Intercept),(Intercept)]             0.006 1.007  554
## mean_PPD                                            0.000 1.000 4162
## log-posterior                                       0.077 1.001 1591
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data111,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
```

```
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 4.659 seconds (Warm-up)
## Chain 1:                4.286 seconds (Sampling)
## Chain 1:                8.945 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 4.402 seconds (Warm-up)
## Chain 2:                3.318 seconds (Sampling)
## Chain 2:                7.72 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: Rejecting initial value:
## Chain 3:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 3:   Stan can't start sampling from this initial value.
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
```

```
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 4.524 seconds (Warm-up)
## Chain 3:                2.867 seconds (Sampling)
## Chain 3:                7.391 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 5.731 seconds (Warm-up)
## Chain 4:                5.389 seconds (Sampling)
## Chain 4:                11.12 seconds (Total)
## Chain 4:


## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 193
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.016  1.021 -3.737 -0.364
## ownership_r_and_rowned                             0.700  0.387  0.051  1.336
## abs(dw_nominate_dim1)                              3.675  1.201  1.702  5.649
## log(pages)                                        -0.285  0.188 -0.609  0.018
## Legislation.typeResolution                         1.890  1.402 -0.096  4.443
## term_has_served                                   -0.032  0.022 -0.069  0.003
## month_since_beginning_congress                     0.012  0.028 -0.035  0.056
## number_of_obs                                      0.006  0.003  0.001  0.012
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.186  0.535 -1.253  0.389
## b[(Intercept) Congress:112th_Congress_(2011-2012)] -0.002  0.378 -0.612  0.587
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.004  0.414 -0.640  0.671
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.055  0.424 -0.779  0.510
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.207  0.468 -0.275  1.141
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.030  0.303 -0.441  0.565
## b[(Intercept) Congress:117th_Congress_(2021-2022)]  0.005  0.406 -0.611  0.664
## Sigma[Congress:(Intercept),(Intercept)]             0.260  0.572  0.000  1.171
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.689  0.041 0.622 0.756
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.016 1.000 4091
## ownership_r_and_rowned                             0.006 1.000 4851
## abs(dw_nominate_dim1)                              0.019 1.000 3939
## log(pages)                                         0.003 0.999 4252
## Legislation.typeResolution                         0.024 1.000 3549
## term_has_served                                    0.000 0.999 3908
## month_since_beginning_congress                     0.000 1.000 4245
## number_of_obs                                      0.000 1.001 2185
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.011 1.001 2537
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.006 1.000 3833
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.007 1.000 3634
```

```
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.007 0.999 3696
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.009 1.000 2611
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.007 1.002 1856
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.012 1.003 1179
## Sigma[Congress:(Intercept),(Intercept)]            0.014 1.000 1617
## mean_PPD                                           0.001 1.000 3683
## log-posterior                                      0.086 1.002 1378
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Table A8

```
data1<-read.csv("C://isq//china bills 2009-2022.csv",fileEncoding="UTF-8-BOM")

data1$dw_nominate_dim1<-as.numeric(data1$dw_nominate_dim1)
```

```
## Warning: NAs introduced by coercion
```

```
data1$pages<-as.numeric(data1$pages)
```

```
## Warning: NAs introduced by coercion
```

```
data1$Legislation.type<-ifelse(data1$Legislation.type=="Bill","Bill","Resolution")
data1$term_has_served<-as.numeric(data1$term_has_served)
data1$month_since_beginning_congress<-as.numeric(data1$month_since_beginning_congress)

data11 <- subset(data1, anti.china %in% c(1, "Cooperation", "Exchanges", "Kindness",
                                          "Trade","Trade Expansion","Immigration Visas"))
nrow(data11) #643
```

```
## [1] 643
```

```
table(data11$Congress)
```

```
##
## 111th Congress (2009-2010) 112th Congress (2011-2012)
##                         45                         48
## 113th Congress (2013-2014) 114th Congress (2015-2016)
##                         37                         27
## 115th Congress (2017-2018) 116th Congress (2019-2020)
##                         53                        170
## 117th Congress (2021-2022)
##                        263
```

```
number_of_obs <- ifelse(data11$Congress == "111th Congress (2009-2010)", 45,
                        ifelse(data11$Congress == "112th Congress (2011-2012)", 48,
                               ifelse(data11$Congress == "113th Congress (2013-2014)", 37,
                                      ifelse(data11$Congress == "114th Congress (2015-2016)", 27,
                                             ifelse(data11$Congress == "115th Congress (2017-2018)", 53
```

```
                                                    ifelse(data11$Congress == "116th Congress (2019-202
                                                           ifelse(data11$Congress == "117th Congress (2

data11$number_of_obs <- as.numeric(number_of_obs)
data111<-subset(data11,data11$policy.area!="International Affairs")
data11$ia<-ifelse(data11$egan_owner_party2=="International Affairs",1,0)
model1<-stan_glmer(formula = complete_partisan~
                     ia+
                     abs(data11$dw_nominate_dim1)+
                     Legislation.type+
                     log(pages)+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data11,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 16.697 seconds (Warm-up)
## Chain 1:                7.933 seconds (Sampling)
## Chain 1:                24.63 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
```

```
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 16.8 seconds (Warm-up)
## Chain 2:                13.281 seconds (Sampling)
## Chain 2:                30.081 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 18.618 seconds (Warm-up)
## Chain 3:                11.608 seconds (Sampling)
## Chain 3:                30.226 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 15.373 seconds (Warm-up)
## Chain 4:                13.648 seconds (Sampling)
## Chain 4:                29.021 seconds (Total)
## Chain 4:
```

```
## Warning: There were 2 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ia + abs(data11$dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 640
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.663  0.519 -3.499 -1.812
## ia                                                -0.700  0.203 -1.043 -0.377
## abs(data11$dw_nominate_dim1)                       3.860  0.599  2.893  4.861
## Legislation.typeResolution                         0.050  0.257 -0.378  0.471
## log(pages)                                        -0.116  0.106 -0.289  0.062
## term_has_served                                   -0.014  0.010 -0.031  0.003
## month_since_beginning_congress                     0.031  0.014  0.009  0.054
## number_of_obs                                      0.008  0.002  0.006  0.011
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.099  0.218 -0.530  0.155
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.082  0.200 -0.160  0.479
## b[(Intercept) Congress:113th_Congress_(2013-2014)] -0.014  0.196 -0.343  0.283
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.020  0.208 -0.372  0.282
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.054  0.190 -0.206  0.397
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.018  0.156 -0.226  0.297
## b[(Intercept) Congress:117th_Congress_(2021-2022)] -0.010  0.215 -0.368  0.335
## Sigma[Congress:(Intercept),(Intercept)]             0.066  0.128  0.000  0.291
```

```
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.523  0.024 0.484 0.562
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.008 1.001 3880
## ia                                                 0.003 1.000 4193
## abs(data11$dw_nominate_dim1)                       0.009 1.001 4600
## Legislation.typeResolution                         0.004 1.000 3978
## log(pages)                                         0.001 1.000 5253
## term_has_served                                    0.000 1.000 4683
## month_since_beginning_congress                     0.000 0.999 3961
## number_of_obs                                      0.000 1.002 1822
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.004 1.000 2611
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.004 1.001 2829
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.003 1.001 4296
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.003 1.000 3989
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.003 1.001 3560
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.004 1.001 1856
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.006 1.001 1328
## Sigma[Congress:(Intercept),(Intercept)]            0.003 1.001 1691
## mean_PPD                                           0.000 1.000 3850
## log-posterior                                      0.083 1.000 1373
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data111,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
```

```
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.871 seconds (Warm-up)
## Chain 1:                5.899 seconds (Sampling)
## Chain 1:                12.77 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 7.69 seconds (Warm-up)
## Chain 2:                6.03 seconds (Sampling)
## Chain 2:                13.72 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: Rejecting initial value:
## Chain 3:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 3:   Stan can't start sampling from this initial value.
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 7.142 seconds (Warm-up)
## Chain 3:                3.848 seconds (Sampling)
## Chain 3:                10.99 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 7.152 seconds (Warm-up)
## Chain 4:                5.125 seconds (Sampling)
## Chain 4:                12.277 seconds (Total)
## Chain 4:

## Warning: There were 3 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
```

```
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 285
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.803  0.813 -4.165 -1.518
## ownership_r_and_rowned                             0.587  0.320  0.061  1.108
## abs(dw_nominate_dim1)                              3.832  0.965  2.262  5.425
## log(pages)                                        -0.293  0.154 -0.545 -0.047
## Legislation.typeResolution                         0.600  0.684 -0.492  1.760
## term_has_served                                   -0.008  0.017 -0.035  0.019
## month_since_beginning_congress                     0.033  0.022 -0.003  0.068
## number_of_obs                                      0.008  0.003  0.004  0.012
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.229  0.423 -1.095  0.221
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.189  0.363 -0.218  0.900
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.090  0.364 -0.405  0.748
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.071  0.389 -0.777  0.461
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.024  0.316 -0.452  0.567
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.058  0.282 -0.354  0.564
## b[(Intercept) Congress:117th_Congress_(2021-2022)] -0.017  0.405 -0.656  0.650
## Sigma[Congress:(Intercept),(Intercept)]            0.209  0.392  0.001  0.893
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.634  0.035 0.575 0.691
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.014 1.000 3281
## ownership_r_and_rowned                             0.005 1.000 4637
## abs(dw_nominate_dim1)                              0.014 1.000 4710
## log(pages)                                         0.002 1.000 4669
## Legislation.typeResolution                         0.010 0.999 4627
## term_has_served                                    0.000 1.001 4496
## month_since_beginning_congress                     0.000 1.001 4121
## number_of_obs                                      0.000 1.002 1402
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.010 1.000 1650
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.010 1.002 1376
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.008 1.001 2259
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.007 1.001 2826
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.006 1.000 2588
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.009 1.003  986
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.014 1.006  811
## Sigma[Congress:(Intercept),(Intercept)]            0.012 1.002 1150
## mean_PPD                                           0.001 1.000 4239
```

```
## log-posterior                                    0.086 1.003 1426
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Table A9

```
data1<-read.csv("C://isq//china bills 2009-2022.csv",fileEncoding="UTF-8-BOM")

data1$dw_nominate_dim1<-as.numeric(data1$dw_nominate_dim1)
```

```
## Warning: NAs introduced by coercion
```

```
data1$pages<-as.numeric(data1$pages)
```

```
## Warning: NAs introduced by coercion
```

```
data1$Legislation.type<-ifelse(data1$Legislation.type=="Bill","Bill","Resolution")
data1$term_has_served<-as.numeric(data1$term_has_served)
data1$month_since_beginning_congress<-as.numeric(data1$month_since_beginning_congress)

data11<-subset(data1,data1$anti.china==1)
nrow(data11)  #603
```

```
## [1] 603
```

```
table(data11$Congress)
```

```
##
## 111th Congress (2009-2010) 112th Congress (2011-2012)
##                         38                         36
## 113th Congress (2013-2014) 114th Congress (2015-2016)
##                         30                         23
## 115th Congress (2017-2018) 116th Congress (2019-2020)
##                         49                        166
## 117th Congress (2021-2022)
##                        261
```

```
number_of_obs <- ifelse(data11$Congress == "111th Congress (2009-2010)", 38,
                        ifelse(data11$Congress == "112th Congress (2011-2012)", 36,
                               ifelse(data11$Congress == "113th Congress (2013-2014)", 30,
                                      ifelse(data11$Congress == "114th Congress (2015-2016)", 23,
                                             ifelse(data11$Congress == "115th Congress (2017-2018)", 49
                                                    ifelse(data11$Congress == "116th Congress (2019-202
                                                           ifelse(data11$Congress == "117th Congress (2


data11$number_of_obs <- as.numeric(number_of_obs)
data111<-subset(data11,data11$policy.area!="International Affairs")
nrow(data111)   #256
```

```
## [1] 256
```

```
data11_0<-subset(data11,data11$Number.of.Cosponsors>0)
data11_0$ia<-ifelse(data11_0$egan_owner_party2=="International Affairs",1,0)
data111_0<-subset(data111,data111$Number.of.Cosponsors>0);nrow(data111_0)
```

```
## [1] 200
```

```
model1<-stan_glmer(formula = complete_partisan~
                     ia+
                     abs(data11_0$dw_nominate_dim1)+
                     Legislation.type+
                     log(pages)+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data11_0,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 11.663 seconds (Warm-up)
## Chain 1:                7.803 seconds (Sampling)
## Chain 1:                19.466 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
```

```
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 5.166 seconds (Warm-up)
## Chain 2:                7.385 seconds (Sampling)
## Chain 2:                12.551 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 11.733 seconds (Warm-up)
## Chain 3:                10.16 seconds (Sampling)
## Chain 3:                21.893 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.01 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 100 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
```

```
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 11.9 seconds (Warm-up)
## Chain 4:                9.166 seconds (Sampling)
## Chain 4:                21.066 seconds (Total)
## Chain 4:
```

```
## Warning: There were 6 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model1,probs=c(0.05,0.95),digits = 3)
```

```
##
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ia + abs(data11_0$dw_nominate_dim1) + Legislation.type +
##     log(pages) + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 489
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.642  0.638 -3.686 -1.609
## ia                                                -0.873  0.239 -1.254 -0.481
## abs(data11_0$dw_nominate_dim1)                     3.932  0.711  2.796  5.111
## Legislation.typeResolution                         0.092  0.299 -0.389  0.580
## log(pages)                                        -0.090  0.129 -0.302  0.124
## term_has_served                                   -0.023  0.013 -0.045 -0.002
## month_since_beginning_congress                     0.013  0.016 -0.013  0.039
## number_of_obs                                      0.007  0.002  0.004  0.010
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.025  0.225 -0.366  0.290
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.031  0.212 -0.258  0.386
## b[(Intercept) Congress:113th_Congress_(2013-2014)] -0.041  0.232 -0.406  0.247
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.015  0.255 -0.331  0.324
## b[(Intercept) Congress:115th_Congress_(2017-2018)] -0.006  0.211 -0.314  0.303
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.024  0.179 -0.227  0.311
```

```
## b[(Intercept) Congress:117th_Congress_(2021-2022)] -0.012  0.225 -0.368  0.315
## Sigma[Congress:(Intercept),(Intercept)]              0.071  0.198  0.000  0.277
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.434  0.027 0.389 0.479
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                    mcse  Rhat  n_eff
## (Intercept)                                        0.014 1.001 2211
## ia                                                 0.004 1.001 4323
## abs(data11_0$dw_nominate_dim1)                     0.011 1.000 4387
## Legislation.typeResolution                         0.005 1.000 4154
## log(pages)                                         0.002 1.000 4036
## term_has_served                                    0.000 1.000 3810
## month_since_beginning_congress                     0.000 0.999 4547
## number_of_obs                                      0.000 1.001 1492
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.008 1.003  776
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.007 1.002  948
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.009 1.004  702
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.008 1.005  922
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.008 1.003  762
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.006 1.005  868
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.008 1.005  892
## Sigma[Congress:(Intercept),(Intercept)]            0.010 1.009  427
## mean_PPD                                           0.000 0.999 4138
## log-posterior                                      0.077 1.002 1528
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
model2<-stan_glmer(formula = complete_partisan~
                     #bill-level
                     ownership_r_and_r+
                     abs(dw_nominate_dim1)+

                     log(pages)+
                     Legislation.type+
                     term_has_served+
                     month_since_beginning_congress+

                     #group-level
                     number_of_obs+
                     (1|Congress),
                   data=data111_0,seed=921,
                   family = binomial(link = "logit"))
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
```

```
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.03 seconds (Warm-up)
## Chain 1:                4.555 seconds (Sampling)
## Chain 1:                10.585 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 5.814 seconds (Warm-up)
## Chain 2:                4.8 seconds (Sampling)
## Chain 2:                10.614 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3: Rejecting initial value:
## Chain 3:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 3:   Stan can't start sampling from this initial value.
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
```

```
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 6.787 seconds (Warm-up)
## Chain 3:                4.831 seconds (Sampling)
## Chain 3:                11.618 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 5.928 seconds (Warm-up)
## Chain 4:                4.842 seconds (Sampling)
## Chain 4:                10.77 seconds (Total)
## Chain 4:

## Warning: There were 2 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
summary(model2,probs=c(0.05,0.95),digits = 3)
```

```
##
```

```
## Model Info:
##  function:     stan_glmer
##  family:       binomial [logit]
##  formula:      complete_partisan ~ ownership_r_and_r + abs(dw_nominate_dim1) +
##     log(pages) + Legislation.type + term_has_served + month_since_beginning_congress +
##     number_of_obs + (1 | Congress)
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 200
##  groups:       Congress (7)
##
## Estimates:
##                                                    mean   sd     5%     95%
## (Intercept)                                       -2.761  1.005 -4.435 -1.178
## ownership_r_and_rowned                             0.428  0.403 -0.236  1.085
## abs(dw_nominate_dim1)                              3.579  1.091  1.781  5.392
## log(pages)                                        -0.356  0.186 -0.665 -0.062
## Legislation.typeResolution                         0.896  0.817 -0.377  2.297
## term_has_served                                   -0.011  0.021 -0.045  0.022
## month_since_beginning_congress                     0.026  0.026 -0.017  0.068
## number_of_obs                                      0.008  0.003  0.003  0.014
## b[(Intercept) Congress:111th_Congress_(2009-2010)] -0.245  0.515 -1.226  0.312
## b[(Intercept) Congress:112th_Congress_(2011-2012)]  0.172  0.453 -0.392  1.080
## b[(Intercept) Congress:113th_Congress_(2013-2014)]  0.095  0.463 -0.557  0.940
## b[(Intercept) Congress:114th_Congress_(2015-2016)] -0.167  0.569 -1.135  0.474
## b[(Intercept) Congress:115th_Congress_(2017-2018)]  0.104  0.414 -0.437  0.887
## b[(Intercept) Congress:116th_Congress_(2019-2020)]  0.133  0.349 -0.335  0.781
## b[(Intercept) Congress:117th_Congress_(2021-2022)] -0.054  0.449 -0.804  0.640
## Sigma[Congress:(Intercept),(Intercept)]             0.319  0.678  0.001  1.275
##
## Fit Diagnostics:
##            mean   sd    5%    95%
## mean_PPD 0.585  0.043 0.510 0.655
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                                   mcse  Rhat  n_eff
## (Intercept)                                       0.017 1.000 3601
## ownership_r_and_rowned                            0.006 0.999 4751
## abs(dw_nominate_dim1)                             0.016 1.000 4744
## log(pages)                                        0.002 1.000 5546
## Legislation.typeResolution                        0.011 0.999 5698
## term_has_served                                   0.000 0.999 5155
## month_since_beginning_congress                    0.000 1.000 5025
## number_of_obs                                     0.000 1.002 1927
## b[(Intercept) Congress:111th_Congress_(2009-2010)] 0.010 1.000 2835
## b[(Intercept) Congress:112th_Congress_(2011-2012)] 0.010 1.001 2126
## b[(Intercept) Congress:113th_Congress_(2013-2014)] 0.009 1.000 2892
## b[(Intercept) Congress:114th_Congress_(2015-2016)] 0.010 1.000 3196
## b[(Intercept) Congress:115th_Congress_(2017-2018)] 0.008 1.002 2452
## b[(Intercept) Congress:116th_Congress_(2019-2020)] 0.009 1.000 1400
## b[(Intercept) Congress:117th_Congress_(2021-2022)] 0.013 1.002 1139
```

```
## Sigma[Congress:(Intercept),(Intercept)]           0.017 1.000 1547
## mean_PPD                                          0.001 1.000 4216
## log-posterior                                     0.086 1.003 1355
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```