```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.72 (7847) x86_64-apple-darwin17.0]

[Workspace restored from /Users/andreasmurr/.RData]
[History restored from /Users/andreasmurr/.Rapp.history]

> setwd('/Users/andreasmurr/Dropbox/citizen-forecasting-germany/replication-archive')
> # Citizen Forecasts of the 2021 German Election
> # Andreas Murr & Mike Lewis-Beck
> # Create Tables 1 and A2 and Figures 1 and A1
>
> # clear working memory
>
>  rm(list=ls())
>
> # ============================
> # = load and write functions =
> # ============================
>
> # load packages
>
>  library(arm)
Loading required package: MASS
Loading required package: Matrix
Loading required package: lme4

arm (Version 1.9-3, built: 2016-11-21)

Working directory is /Users/andreasmurr/Dropbox/citizen-forecasting-germany/replication-
archive

>  library(systemfit)
Loading required package: car
Loading required package: carData
Registered S3 methods overwritten by 'car':
  method                          from
  influence.merMod                lme4
  cooks.distance.influence.merMod lme4
  dfbeta.influence.merMod         lme4
  dfbetas.influence.merMod        lme4

Attaching package: ‘car’
```

```
The following object is masked from ‘package:arm’:

    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


Please cite the 'systemfit' package as:
Arne Henningsen and Jeff D. Hamann (2007). systemfit: A Package for Estimating Systems of
Simultaneous Equations in R. Journal of Statistical Software 23(4), 1-40. http://
www.jstatsoft.org/v23/i04/.

If you have questions, suggestions, or comments regarding the 'systemfit' package, please use
a forum or 'tracker' at systemfit's R-Forge site:
https://r-forge.r-project.org/projects/systemfit/
>
> # write functions
>
>  loo.errors = function(x){
+    e = x$residuals
+    h = lm.influence(x)$hat
+    sqrt(mean((e / (1 - h))^2))
+ }
>
>  y.hat.i = function(fit){
+    e = residuals(fit)
+    y.hat = fitted(fit)
+    h = lm.influence(fit)$hat
+    y.hat.i = y.hat - h * e / (1 - h)

+ }
>
>  sum.na = function(x){sum(x, na.rm=TRUE)}
>
>  normalise = function(x){
+    apply(x, 1, function(r){
+       s = sum.na(r)
+       if (s>1) {
+          r / s
+       } else {
+          r
+       }
+    })
+ }
>
>  rmse.na = function(x,y){
+    sqrt(mean((x-y)^2, na.rm=TRUE))
+ }
>
> # ==============
> # = manuscript =
> # ==============
>
> # load data
```

```
>
>  data2 = read.csv("data2.csv", header=T)
>
> #--------
> # table 1
> #--------
>
> # fit models
>
>  m.cdu = lm(vote.cdu ~ exp.cdu + gcoal, data=data2)
>  m.spd = lm(vote.spd ~ exp.spd + gcoal, data=data2)
>  m.fdp = lm(vote.fdp ~ exp.fdp + gcoal, data=data2)
>  m.gru = lm(vote.gru ~ exp.gru + gcoal, data=data2)
>  m.lin = lm(vote.lin ~ exp.lin + gcoal, data=data2)
>
> # display models
>
>  display(m.cdu)
lm(formula = vote.cdu ~ exp.cdu + gcoal, data = data2)
            coef.est coef.se
(Intercept)  0.31     0.05
exp.cdu      0.20     0.09
gcoal       -0.09     0.03
---
n = 10, k = 3
residual sd = 0.04, R-Squared = 0.64
>  display(m.spd)
lm(formula = vote.spd ~ exp.spd + gcoal, data = data2)
            coef.est coef.se
(Intercept)  0.31     0.02
exp.spd      0.22     0.07
gcoal       -0.10     0.03
---
n = 10, k = 3
residual sd = 0.03, R-Squared = 0.86
>  display(m.fdp)
lm(formula = vote.fdp ~ exp.fdp + gcoal, data = data2)
            coef.est coef.se
(Intercept) 0.06     0.01
exp.fdp     0.25     0.10
gcoal       0.05     0.01
---
n = 10, k = 3
residual sd = 0.02, R-Squared = 0.74
>  display(m.gru)
lm(formula = vote.gru ~ exp.gru + gcoal, data = data2)
            coef.est coef.se
(Intercept) 0.06     0.01
exp.gru     0.60     0.26
gcoal       0.04     0.01
---
n = 10, k = 3
residual sd = 0.01, R-Squared = 0.64
>  display(m.lin)
lm(formula = vote.lin ~ exp.lin + gcoal, data = data2)
            coef.est coef.se
(Intercept) 0.04     0.01
exp.lin     8.09     2.10
gcoal       0.04     0.01
---
n = 8, k = 3
```

```
residual sd = 0.01, R-Squared = 0.88
>
> # compute out-of-sample errors
>
>  round(loo.errors(m.cdu), 2)
[1] 0.04
>  round(loo.errors(m.spd), 2)
[1] 0.05
>  round(loo.errors(m.fdp), 2)
[1] 0.02
>  round(loo.errors(m.gru), 2)
[1] 0.02
>  round(loo.errors(m.lin), 2)
[1] 0.02
>
> # compute normalised leave-one-out-error
>
>  Y.hat = cbind(y.hat.i(m.cdu), y.hat.i(m.spd), y.hat.i(m.fdp), y.hat.i(m.gru),
c(NA,NA,y.hat.i(m.lin)))
>  Y.hat.norm = t(normalise(Y.hat))
>  Y.hat.norm = data.frame(Y.hat.norm)
>  names(Y.hat.norm) = c("cdu", "spd", "fdp", "gru", "lin")
>  Y.hat.norm$oth = 1 - apply(Y.hat.norm, 1, sum.na)
>
>  round(rmse.na(data2$vote.cdu, Y.hat.norm$cdu), 2)
[1] 0.03
>  round(rmse.na(data2$vote.spd, Y.hat.norm$spd), 2)
[1] 0.04
>  round(rmse.na(data2$vote.fdp, Y.hat.norm$fdp), 2)
[1] 0.02
>  round(rmse.na(data2$vote.gru, Y.hat.norm$gru), 2)
[1] 0.02
>  round(rmse.na(data2$vote.lin, Y.hat.norm$lin), 2)
[1] 0.02
>  round(rmse.na(data2$vote.oth, Y.hat.norm$oth), 2)
[1] 0.06
>
> # predict chancellor
>
>  data2$year[apply(Y.hat.norm, 1, which.max)!=data2$chancellor]
[1] 2002
>  mean(apply(Y.hat.norm, 1, which.max)==data2$chancellor)
[1] 0.9
>
> #-------------------------
> # equality of coefficients
> #-------------------------
>
>  equation.system = list(
+    cdu = vote.cdu ~ exp.cdu + gcoal,
+    spd = vote.spd ~ exp.spd + gcoal,
+    fdp = vote.fdp ~ exp.fdp + gcoal,
+    gru = vote.gru ~ exp.gru + gcoal
+ )
>  fit = systemfit( equation.system , data=data2, method="SUR")
>  restrict = c("cdu_exp.cdu - spd_exp.spd = 0", "cdu_exp.cdu - fdp_exp.fdp = 0", "cdu_exp.cdu
- gru_exp.gru = 0")
>  linearHypothesis(fit, hypothesis.matrix = restrict)
Linear hypothesis test (Theil's F test)

Hypothesis:
```

```
cdu_exp.cdu - spd_exp.spd = 0
cdu_exp.cdu - fdp_exp.fdp = 0
cdu_exp.cdu - gru_exp.gru = 0

Model 1: restricted model
Model 2: fit

  Res.Df Df      F Pr(>F)
1     31
2     28  3 0.7953 0.5069
>
> #---------
> # figure 1
> #---------
>
>  # set graphical parameters
>  par(las=1,mar=c(3.2,3.2,2,1), mgp=c(2.2,.7,0), tck=-.01, mfrow=c(3,2))

>  x.lim = range(c(data2$exp.cdu, data2$exp.spd, data2$exp.fdp, data2$exp.gru,
data2$exp.lin[data2$year>=1990]))
>  y.lim = range(c(data2$vote.cdu, data2$vote.spd, data2$vote.fdp, data2$vote.gru,
data2$vote.lin[data2$year>=1990]))
>  # cdu/csu
>  with(data2, plot(exp.cdu, vote.cdu, col=ifelse(gcoal==0, "black", "darkgrey"), pch=16,
xlim=x.lim, ylim=y.lim, xlab="Expectations", ylab="Vote share", main="CDU/CSU"))
>  curve(cbind(1,x,0)%*%coef(m.cdu), lwd=1, col="black", add=TRUE)
>  curve(cbind(1,x,1)%*%coef(m.cdu), lwd=1, col="darkgrey", add=TRUE)
>  # spd
>  with(data2, plot(exp.spd, vote.spd, col=ifelse(gcoal==0, "black", "darkgrey"), pch=16,
xlim=x.lim, ylim=y.lim, xlab="Expectations", ylab="Vote share", main="SPD"))
>  curve(cbind(1,x,0)%*%coef(m.spd), lwd=1, col="black", add=TRUE)
>  curve(cbind(1,x,1)%*%coef(m.spd), lwd=1, col="darkgrey", add=TRUE)
>  # fdp
>  with(data2, plot(exp.fdp, vote.fdp, col=ifelse(gcoal==0, "black", "darkgrey"), pch=16,
xlim=x.lim, ylim=y.lim, xlab="Expectations", ylab="Vote share", main="FDP"))
>  curve(cbind(1,x,0)%*%coef(m.fdp), lwd=1, col="black", add=TRUE)
>  curve(cbind(1,x,1)%*%coef(m.fdp), lwd=1, col="darkgrey", add=TRUE)
>  # gru
>  with(data2, plot(exp.gru, vote.gru, col=ifelse(gcoal==0, "black", "darkgrey"), pch=16,
xlim=x.lim, ylim=y.lim, xlab="Expectations", ylab="Vote share", main="Grüne"))
>  curve(cbind(1,x,0)%*%coef(m.gru), lwd=1, col="black", add=TRUE)
>  curve(cbind(1,x,1)%*%coef(m.gru), lwd=1, col="darkgrey", add=TRUE)
>  # linke
>  with(data2[data2$year>=1990,], plot(exp.lin, vote.lin, col=ifelse(gcoal==0, "black",
"darkgrey"), xlim=x.lim, ylim=y.lim, pch=16, xlab="Expectations", ylab="Vote share",
main="Linke"))
>  curve(cbind(1,x,0)%*%coef(m.lin), lwd=1, col="black", add=TRUE)
>  curve(cbind(1,x,1)%*%coef(m.lin), lwd=1, col="darkgrey", add=TRUE)
>  # linke (zoom)
>  with(data2[data2$year>=1990,], plot(exp.lin, vote.lin, col=ifelse(gcoal==0, "black",
"darkgrey"), pch=16, xlab="Expectations", ylab="Vote share", main="Linke (Zoom)"))
>  curve(cbind(1,x,0)%*%coef(m.lin), lwd=1, col="black", add=TRUE)
>  curve(cbind(1,x,1)%*%coef(m.lin), lwd=1, col="darkgrey", add=TRUE)
>
> #---------
> # forecast
> #---------
>
>  newdata = data.frame(exp.cdu=.64, exp.spd=.03, exp.gru=.09, gcoal=1)
>  y.hat.cdu = predict(m.cdu, newdata)
>  y.hat.spd = predict(m.spd, newdata)
```

```
> y.hat.gru = predict(m.gru, newdata)
>
> round(y.hat.cdu, 2)
   1
0.34
> round(y.hat.spd, 2)
   1
0.21
> round(y.hat.cdu + y.hat.spd, 2)
   1
0.55
> round(y.hat.gru, 2)
   1
0.15
> round(y.hat.cdu + y.hat.gru, 2)
   1
0.49
>
> # ====================
> # = online appendix =
> # ====================
>
> #---------
> # table a2
> #---------
>
> # please see create-data.R
>
> #---------
> # table a2
> #---------
>
> # load data
>
> data1 = read.csv("data1.csv", header=T)
> data3 = read.csv("data3.csv", header=T)
> data5 = read.csv("data5.csv", header=T)
>
> # fit models
>
> # 2 months on various election sets
> m.cdu.2 = lm(vote.cdu ~ exp.cdu + gcoal, data=data2)
> m.spd.2 = lm(vote.spd ~ exp.spd + gcoal, data=data2)
> m.fdp.2 = lm(vote.fdp ~ exp.fdp + gcoal, data=data2)
> m.gru.2 = lm(vote.gru ~ exp.gru + gcoal, data=data2)
> m.lin.2 = lm(vote.lin ~ exp.lin + gcoal, data=data2)
> m.cdu.2.1 = lm(vote.cdu ~ exp.cdu + gcoal, data=data2[data2$year%in%data1$year,])
> m.spd.2.1 = lm(vote.spd ~ exp.spd + gcoal, data=data2[data2$year%in%data1$year,])
> m.fdp.2.1 = lm(vote.fdp ~ exp.fdp + gcoal, data=data2[data2$year%in%data1$year,])
> m.gru.2.1 = lm(vote.gru ~ exp.gru + gcoal, data=data2[data2$year%in%data1$year,])
> m.lin.2.1 = lm(vote.lin ~ exp.lin + gcoal, data=data2[data2$year%in%data1$year,])
> m.cdu.2.3 = lm(vote.cdu ~ exp.cdu + gcoal, data=data2[data2$year%in%data3$year,])
> m.spd.2.3 = lm(vote.spd ~ exp.spd + gcoal, data=data2[data2$year%in%data3$year,])
> m.fdp.2.3 = lm(vote.fdp ~ exp.fdp + gcoal, data=data2[data2$year%in%data3$year,])
> m.gru.2.3 = lm(vote.gru ~ exp.gru + gcoal, data=data2[data2$year%in%data3$year,])
> m.lin.2.3 = lm(vote.lin ~ exp.lin + gcoal, data=data2[data2$year%in%data3$year,])
> m.cdu.2.5 = lm(vote.cdu ~ exp.cdu + gcoal, data=data2[data2$year%in%data5$year,])
> m.spd.2.5 = lm(vote.spd ~ exp.spd + gcoal, data=data2[data2$year%in%data5$year,])
> m.fdp.2.5 = lm(vote.fdp ~ exp.fdp + gcoal, data=data2[data2$year%in%data5$year,])
> m.gru.2.5 = lm(vote.gru ~ exp.gru + gcoal, data=data2[data2$year%in%data5$year,])
> m.lin.2.5 = lm(vote.lin ~ exp.lin + gcoal, data=data2[data2$year%in%data5$year,])
```

```
>  # 1 month
>  m.cdu.1 = lm(vote.cdu ~ exp.cdu + gcoal, data=data1)
>  m.spd.1 = lm(vote.spd ~ exp.spd + gcoal, data=data1)
>  m.fdp.1 = lm(vote.fdp ~ exp.fdp + gcoal, data=data1)
>  m.gru.1 = lm(vote.gru ~ exp.gru + gcoal, data=data1)
>  m.lin.1 = lm(vote.lin ~ exp.lin + gcoal, data=data1)
>  # 3 months
>  m.cdu.3 = lm(vote.cdu ~ exp.cdu + gcoal, data=data3)
>  m.spd.3 = lm(vote.spd ~ exp.spd + gcoal, data=data3)
>  m.fdp.3 = lm(vote.fdp ~ exp.fdp + gcoal, data=data3)
>  m.gru.3 = lm(vote.gru ~ exp.gru + gcoal, data=data3)
>  m.lin.3 = lm(vote.lin ~ exp.lin + gcoal, data=data3)
>  # 5 months
>  m.cdu.5 = lm(vote.cdu ~ exp.cdu + gcoal, data=data5)
>  m.spd.5 = lm(vote.spd ~ exp.spd + gcoal, data=data5)
>  m.fdp.5 = lm(vote.fdp ~ exp.fdp + gcoal, data=data5)
>  m.gru.5 = lm(vote.gru ~ exp.gru + gcoal, data=data5)
>  m.lin.5 = lm(vote.lin ~ exp.lin + gcoal, data=data5)
>
> # normalised loocv error
>
>  Y.hat = cbind(y.hat.i(m.cdu.2), y.hat.i(m.spd.2), y.hat.i(m.fdp.2), y.hat.i(m.gru.2),
c(rep(NA, length(y.hat.i(m.gru.2)) - length(y.hat.i(m.lin.2))), y.hat.i(m.lin.2)))
>  Y.hat.2.1 = cbind(y.hat.i(m.cdu.2.1), y.hat.i(m.spd.2.1), y.hat.i(m.fdp.2.1),
y.hat.i(m.gru.2.1), c(rep(NA, length(y.hat.i(m.gru.2.1)) - length(y.hat.i(m.lin.2.1))),
y.hat.i(m.lin.2.1)))
>  Y.hat.2.3 = cbind(y.hat.i(m.cdu.2.3), y.hat.i(m.spd.2.3), y.hat.i(m.fdp.2.3),
y.hat.i(m.gru.2.3), c(rep(NA, length(y.hat.i(m.gru.2.3)) - length(y.hat.i(m.lin.2.3))),
y.hat.i(m.lin.2.3)))
>  Y.hat.2.5 = cbind(y.hat.i(m.cdu.2.5), y.hat.i(m.spd.2.5), y.hat.i(m.fdp.2.5),
y.hat.i(m.gru.2.5), c(rep(NA, length(y.hat.i(m.gru.2.5)) - length(y.hat.i(m.lin.2.5))),
y.hat.i(m.lin.2.5)))
>  Y.hat.1 = cbind(y.hat.i(m.cdu.1), y.hat.i(m.spd.1), y.hat.i(m.fdp.1), y.hat.i(m.gru.1),
c(rep(NA, length(y.hat.i(m.gru.1)) - length(y.hat.i(m.lin.1))), y.hat.i(m.lin.1)))
>  Y.hat.3 = cbind(y.hat.i(m.cdu.3), y.hat.i(m.spd.3), y.hat.i(m.fdp.3), y.hat.i(m.gru.3),
c(rep(NA, length(y.hat.i(m.gru.3)) - length(y.hat.i(m.lin.3))), y.hat.i(m.lin.3)))
>  Y.hat.5 = cbind(y.hat.i(m.cdu.5), y.hat.i(m.spd.5), y.hat.i(m.fdp.5), y.hat.i(m.gru.5),
c(rep(NA, length(y.hat.i(m.gru.5)) - length(y.hat.i(m.lin.5))), y.hat.i(m.lin.5)))
>
>  Y.hat.norm = t(normalise(Y.hat))
>  Y.hat.norm.2.1 = t(normalise(Y.hat.2.1))
>  Y.hat.norm.2.3 = t(normalise(Y.hat.2.3))
>  Y.hat.norm.2.5 = t(normalise(Y.hat.2.5))
>  Y.hat.norm.1 = t(normalise(Y.hat.1))
>  Y.hat.norm.3 = t(normalise(Y.hat.3))
>  Y.hat.norm.5 = t(normalise(Y.hat.5))
>
>  Y.hat.norm = data.frame(Y.hat.norm)
>  Y.hat.norm.2.1 = data.frame(Y.hat.norm.2.1)
>  Y.hat.norm.2.3 = data.frame(Y.hat.norm.2.3)
>  Y.hat.norm.2.5 = data.frame(Y.hat.norm.2.5)
>  Y.hat.norm.1 = data.frame(Y.hat.norm.1)
>  Y.hat.norm.3 = data.frame(Y.hat.norm.3)
>  Y.hat.norm.5 = data.frame(Y.hat.norm.5)
>
>  names(Y.hat.norm) = names(Y.hat.norm.2.1) = names(Y.hat.norm.2.3) = names(Y.hat.norm.2.5) =
names(Y.hat.norm.1) = names(Y.hat.norm.3) = names(Y.hat.norm.5) = c("cdu", "spd", "fdp",
"gru", "lin")
>  Y.hat.norm$oth = 1 - apply(Y.hat.norm, 1, sum.na)
>  Y.hat.norm.2.1$oth = 1 - apply(Y.hat.norm.2.1, 1, sum.na)
>  Y.hat.norm.2.3$oth = 1 - apply(Y.hat.norm.2.3, 1, sum.na)
```

```
> Y.hat.norm.2.5$oth = 1 - apply(Y.hat.norm.2.5, 1, sum.na)
> Y.hat.norm.1$oth = 1 - apply(Y.hat.norm.1, 1, sum.na)
> Y.hat.norm.3$oth = 1 - apply(Y.hat.norm.3, 1, sum.na)
> Y.hat.norm.5$oth = 1 - apply(Y.hat.norm.5, 1, sum.na)
>
> loo.norm = c(
+    rmse.na(data2$vote.cdu, Y.hat.norm$cdu),
+    rmse.na(data2$vote.spd, Y.hat.norm$spd),
+    rmse.na(data2$vote.fdp, Y.hat.norm$fdp),
+    rmse.na(data2$vote.gru, Y.hat.norm$gru),
+    rmse.na(data2$vote.lin, Y.hat.norm$lin),
+    rmse.na(data2$vote.oth, Y.hat.norm$oth)
+ )
> loo.norm.2.1 = c(
+    rmse.na(data1$vote.cdu, Y.hat.norm.2.1$cdu),
+    rmse.na(data1$vote.spd, Y.hat.norm.2.1$spd),
+    rmse.na(data1$vote.fdp, Y.hat.norm.2.1$fdp),
+    rmse.na(data1$vote.gru, Y.hat.norm.2.1$gru),
+    rmse.na(data1$vote.lin, Y.hat.norm.2.1$lin),
+    rmse.na(data1$vote.oth, Y.hat.norm.2.1$oth)
+ )
> loo.norm.2.3 = c(
+    rmse.na(data3$vote.cdu, Y.hat.norm.2.3$cdu),
+    rmse.na(data3$vote.spd, Y.hat.norm.2.3$spd),
+    rmse.na(data3$vote.fdp, Y.hat.norm.2.3$fdp),
+    rmse.na(data3$vote.gru, Y.hat.norm.2.3$gru),
+    rmse.na(data3$vote.lin, Y.hat.norm.2.3$lin),
+    rmse.na(data3$vote.oth, Y.hat.norm.2.3$oth)
+ )
> loo.norm.2.5 = c(
+    rmse.na(data5$vote.cdu, Y.hat.norm.2.5$cdu),
+    rmse.na(data5$vote.spd, Y.hat.norm.2.5$spd),
+    rmse.na(data5$vote.fdp, Y.hat.norm.2.5$fdp),
+    rmse.na(data5$vote.gru, Y.hat.norm.2.5$gru),
+    rmse.na(data5$vote.lin, Y.hat.norm.2.5$lin),
+    rmse.na(data5$vote.oth, Y.hat.norm.2.5$oth)
+ )
> loo.norm.1 = c(
+    rmse.na(data1$vote.cdu, Y.hat.norm.1$cdu),
+    rmse.na(data1$vote.spd, Y.hat.norm.1$spd),
+    rmse.na(data1$vote.fdp, Y.hat.norm.1$fdp),
+    rmse.na(data1$vote.gru, Y.hat.norm.1$gru),
+    rmse.na(data1$vote.lin, Y.hat.norm.1$lin),
+    rmse.na(data1$vote.oth, Y.hat.norm.1$oth)
+ )
> loo.norm.3 = c(
+    rmse.na(data3$vote.cdu, Y.hat.norm.3$cdu),
+    rmse.na(data3$vote.spd, Y.hat.norm.3$spd),
+    rmse.na(data3$vote.fdp, Y.hat.norm.3$fdp),
+    rmse.na(data3$vote.gru, Y.hat.norm.3$gru),
+    rmse.na(data3$vote.lin, Y.hat.norm.3$lin),
+    rmse.na(data3$vote.oth, Y.hat.norm.3$oth)
+ )
> loo.norm.5 = c(
+    rmse.na(data5$vote.cdu, Y.hat.norm.5$cdu),
+    rmse.na(data5$vote.spd, Y.hat.norm.5$spd),
+    rmse.na(data5$vote.fdp, Y.hat.norm.5$fdp),
+    rmse.na(data5$vote.gru, Y.hat.norm.5$gru),
+    rmse.na(data5$vote.lin, Y.hat.norm.5$lin),
+    rmse.na(data5$vote.oth, Y.hat.norm.5$oth)
+ )
```

```
>
>  loo.norm.tab = rbind(
+    cbind(loo.norm, loo.norm.2.1, loo.norm.1, loo.norm.2.3, loo.norm.3, loo.norm.2.5,
loo.norm.5),
+    c(mean(loo.norm), mean(loo.norm.2.1), mean(loo.norm.1), mean(loo.norm.2.3),
mean(loo.norm.3), mean(loo.norm.2.5), mean(loo.norm.5))
+  )
>
>  round(loo.norm.tab*100, 1)
     loo.norm loo.norm.2.1 loo.norm.1 loo.norm.2.3 loo.norm.3 loo.norm.2.5 loo.norm.5
[1,]      3.4          3.4        3.7          3.8        5.5          4.3        6.5
[2,]      4.5          5.6        3.9          2.3        1.5          2.4        1.9
[3,]      1.8          2.4        2.0          1.0        1.7          1.1        1.9
[4,]      1.9          1.1        1.4          1.9        1.8          1.8        2.0
[5,]      1.7          1.7        4.0          1.7        1.4          1.7        2.7
[6,]      6.2          7.3        5.5          5.2        5.2          5.5        6.9
[7,]      3.2          3.6        3.4          2.6        2.9          2.8        3.7
>
> #----------
> # figure a1
> #----------
>
>  par(las=1,mar=c(3.2,3.2,2,1), mgp=c(2.2,.7,0), tck=-.01, mfrow=c(3,2))

>  plot(data2$vote.cdu, Y.hat.norm$cdu, xlim=0:1, ylim=0:1, xlab="Actual", ylab="Predicted",
main="CDU/CSU")
>  abline(a=0,b=1, lty=2)
>  plot(data2$vote.spd, Y.hat.norm$spd, xlim=0:1, ylim=0:1, xlab="Actual", ylab="Predicted",
main="SPD")
>  abline(a=0,b=1, lty=2)
>  plot(data2$vote.fdp, Y.hat.norm$fdp, xlim=0:1, ylim=0:1, xlab="Actual", ylab="Predicted",
main="FDP")
>  abline(a=0,b=1, lty=2)
>  plot(data2$vote.gru, Y.hat.norm$gru, xlim=0:1, ylim=0:1, xlab="Actual", ylab="Predicted",
main="Grüne")
>  abline(a=0,b=1, lty=2)
>  plot(data2$vote.lin, Y.hat.norm$lin, xlim=0:1, ylim=0:1, xlab="Actual", ylab="Predicted",
main="Linke")
>  abline(a=0,b=1, lty=2)
>  plot(data2$vote.oth, Y.hat.norm$oth, xlim=0:1, ylim=0:1, xlab="Actual", ylab="Predicted",
main="Others")
>  abline(a=0,b=1, lty=2)
>
> # ====================
> # = end source code =
> # ====================
>
```