

Agregador_Formulas

Natalia Tosi

10/16/2021

1. Functions

```
#source("R/Extract.r") #load stimsons "extract" function (downloaded from internet)
source("https://raw.githubusercontent.com/nataliatosi/nataliatosi/main/Agregador_BR/R/Extract.r")

observations <- wcalcdiagnosticsQ <- wcalcdiagnosticsM <- list()
```

1.1. Function to identify months that used data from more than one president

```
my.drop <- function(xx){
  xx$drop.wcalcM <- xx$drop.wcalcQ <- F #indicator for whether to drop
  #Drop months after last obs and prior to first obs
  for (i in levels(xx$PresidentS)){
    firstmonth <- min(which(xx$PresidentS==i))
    firstpop <- min(which(is.na(xx$Positive)==F&xx$PresidentS==i))
    lastmonth <- max(which(xx$PresidentS==i))
    lastpop <- max(which(is.na(xx$Positive)==F&xx$PresidentS==i))
    if(firstmonth<firstpop){xx$drop.wcalcM[firstmonth:(firstpop-1)] <- T}
    if(lastpop<lastmonth){xx$drop.wcalcM[(lastpop+1):lastmonth] <- T}
  }
  months.to.drop <- xx$M[xx$drop.wcalcM] #this is final
  #For quarters, drop after last obs, prior to first obs and
  #those with two presidents
  Qpost <- unique(xx$Q[xx$drop.wcalcM]) #refine this later
  tmp <- dapply(xx,.(Q),function(x){sum(is.na(x$Positive))})
  QNA <- names(tmp[tmp==3]) #quarters with no data
  tmp <- dapply(xx,.(Q),function(x){length(unique(x$PresidentS[is.na(x$Positive)==F]))})
  Q2P <- names(tmp[tmp==2]) #quarters With DATA from two presidents
  quarters.do.drop <- union(intersect(Qpost,QNA),Q2P)
  #out <- list(M=months.to.drop,Q=quarters.do.drop)
  xx$drop.wcalcQ[is.element(xx$Q,quarters.do.drop)] <- T
  ## The line below might seem strange, but it doesn't make sense to keep
  ## monthly observations of Q for months that have been dropped
  ## this is specially a problem for transition between presidents
  xx$drop.wcalcQ[is.element(xx$M,months.to.drop)] <- T
  cat('Should drop WCALC estimates for the following months:\n')
  print(months.to.drop)
```

```

cat('and quarters:\n')
print(quarters.do.drop)
return(xx)
}

```

1.2. Functions for aggregating monthly data

```

obsM <- function(d){nrow(d)}
institutesM <- function(d){length(unique(d$Institute))}
instituteM <- function(d){if(length(unique(d$Institute))==1){
  as.character(d$Institute[1])}else{"_Multiple"}}

my.averageM <- function(x,drop.series=NULL){
  if(is.null(drop.series)==F){ x <- subset(x,Institute!=drop.series)}
  tmp <- dply(x, "M", c("popM","obsM","institutesM"))
  tmp2 <- dply(x, "M", c("instituteM","presUsed"))
  out <- merge(tmp,tmp2,by="M",all=T)
  return(out)
}

popM <- function(d){#simple aggregation
  ifelse(is.nan(mean(d$Positive,na.rm=T)),NA,mean(d$Positive,na.rm=T))}
popM <- function(x){#for when there is more than one president in the same term
  if(length(unique(x$PresidentS))==1){
    ifelse(is.nan(mean(x$Positive,na.rm=T)),NA,mean(x$Positive,na.rm=T))
  }else{#if more than one president, use incoming
    to.keep <- which(x$PresidentS==x$PresidentS[nrow(x)])
    ifelse(is.nan(mean(x$Positive[to.keep],na.rm=T)),NA,mean(x$Positive[to.keep],na.rm=T))
  }}

presUsed <- function(d){if(length(unique(d$PresidentS))==1){#use incoming president
  as.character(d$PresidentS[1])}else{as.character(d$PresidentS[nrow(d)])}}

dateUsed <- function(d){ #if no observation, use start of quarter,
  if(is.na(max(d$date))){
    out <- gsub("-1","-02-15",d$Q)
    out <- gsub("-2","-05-15",out)
    out <- gsub("-3","-08-15",out)
    out <- gsub("-4","-11-15",out)
  }else{
    out <- max(d$date)
  }
  return(out)}

dateUsed <- function(d){max(d$date)}

my.averageQ <- function(x,drop.series=NULL){
  if(is.null(drop.series)==F){ x <- subset(x,Institute!=drop.series)}
  tmp <- dply(x, "Q", c("popM",
    "obsM","institutesM"))
  tmp2 <- dply(x, "Q", c("instituteM","presUsed"))

```

```

tmp3 <- ddply(x, "Q", c("dateUsed"))
out <- merge(merge(tmp, tmp2, by="Q", all=T), tmp3, by="Q", all=T)
names(out) <- gsub("M", "Q", names(out))
return(out)
}

```

2. Setting

2.1. Create empty dataframe for merging results later

```

#### Merge estimates into d dataset for plotting ####
Ms <- expand.grid(1980:2017, sprintf("%02.0f", 1:12))
Ms <- sort(paste(Ms[,1], Ms[,2], sep="-"))
Qs <- gsub("-01|-02|-03", "-1"
          , gsub("-04|-05|-06", "-2"
                , gsub("-07|-08|-09", "-3"
                      , gsub("-10|-11|-12", "-4", Ms))))
MQ <- data.frame(M=Ms, Q=Qs)

#define a month
m1 <- 365/12

```

3. Prepare data from stimsons' wcalc

```

#load("R/popularity_raw_BR.RData")

#d <- read_sheet("https://docs.google.com/spreadsheets/d/1c9TIhLOhUUpRCdKeVv7mMzGNuM7YpyJ_2ffmv-bhHrE/e")

d <- read_csv(url("https://raw.githubusercontent.com/nataliatosi/nataliatosi/main/Agregador_BR/DATA/popularity_raw_BR.RData"))

## Warning: Missing column names filled in: 'X1' [1]

##
## -- Column specification -----
## cols(
##   .default = col_character(),
##   X1 = col_double(),
##   year = col_double(),
##   Positive = col_double(),
##   Negative = col_double(),
##   'DK-Neutral' = col_double(),
##   Neutral = col_double(),
##   DK = col_double(),
##   SampleSize = col_double(),
##   date = col_date(format = "")
## )
## i Use 'spec()' for the full column specifications.

```

```

### The simple averaging approach #####
ms <- my.averageM(d) #by month
qs <- my.averageQ(d) #by quarter

#For WCalc, start by saving the info of which pres to use into data
#this is to make sure only "incoming" presidents are used when
#there are more than one per time period
#this should not affect monthly latent estimates
#but definitely affects quarterly

d <- merge(d, subset(ms, select = c(M,presUsed)), by = "M", all.x = T)
d$useM <- d$PresidentS == d$presUsed

d$Varname <- gsub("\\s","", d$Institute)
d$Date <- d$date
d$Index <- d$Positive

ds <- subset(d, select = c(Varname,Date,Index,PresidentS,Q,M,useM))

cat("Observation by pollster in dataset\n")

```

```
## Observation by pollster in dataset
```

```
print(table(ds$Varname))
```

```
##
##      Atlas      Datafolha DataPoder360      Gallup      IBOPE      IBPAD
##      14         205         40          63         171         7
## IdeiaBigData      IPESPE      IPSOS      MDA      Offerwise      Parana
##      71         40         45         30         6         5
##      Quaest      Sensus      Vox
##      7         76         46
```

3.1. WCALC Monthly

```
useM <- which(ds$useM == T)
```

```

#wcalc.Mraw <- extract(varname = ds$Varname[useM], date = ds$Date[useM], index = ds$Index[useM], unit =
wcalc.Mraw <- extract(varname = ds$Varname[useM],
                      date = ds$Date[useM],
                      index = ds$Index[useM],
                      unit = "M")

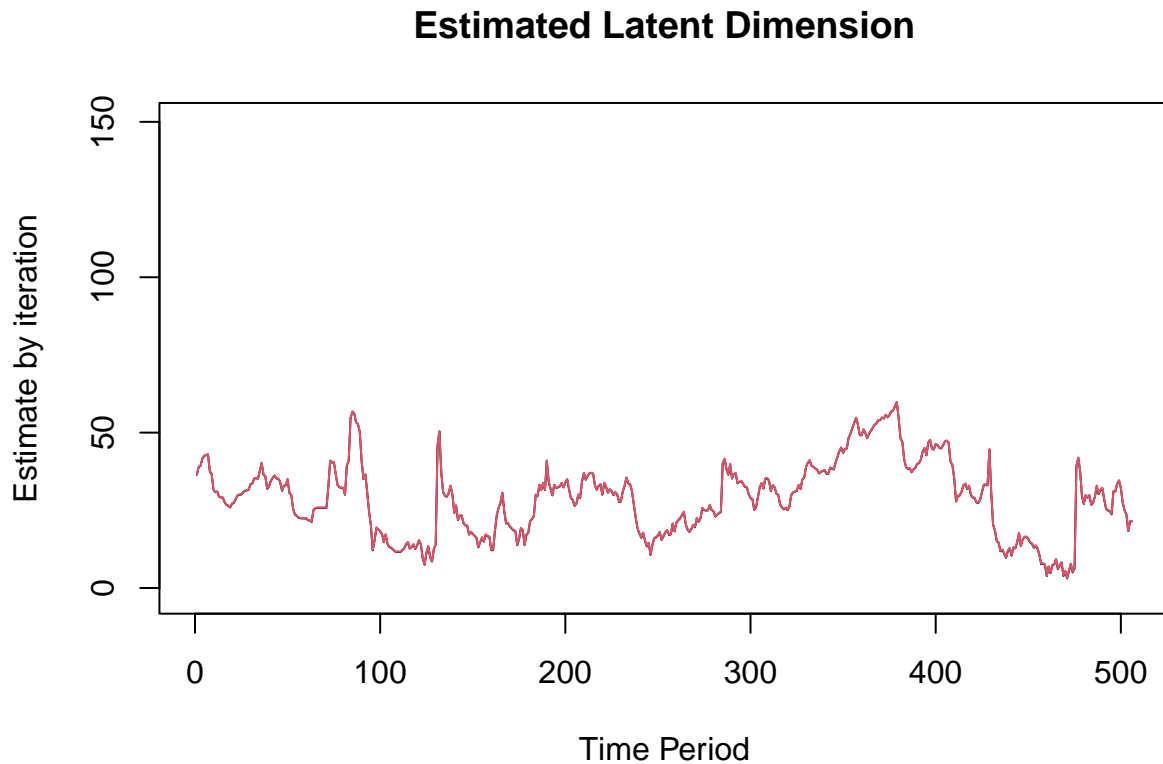
```

```

## [1] "Estimation report:"
## [1] "Period: 1979 5 to 2021 6 506 time points"
## [1] "Number of series: 15"
## [1] "Number of usable series: 15"
## [1] "Exponential smoothing: TRUE"
## [1] "Iteration history: Dimension 1"

```

```
## [1] " "
## [1] "Iter Convergence Criterion Reliability Alphaf Alphab"
```



```
## [1] "1          0.0168      0.001      0.934 0.7286 0.8317"
## [1] "2          0.0023      0.001      0.934 0.728 0.8343"
## [1] "3          3e-04      0.001      0.935 0.7278 0.8344"
## [1] " "
## [1] "Eigen Estimate 1.27 of possible 1.35"
## [1] " Percent Variance Explained: 93.57"
## [1] " "
## [1] "Final Weighted Average Metric: Mean: 28.97 St. Dev: 11.85"
```

```
wcalcdiagnosticsM[["brazil"]] <- summary(wcalc.Mraw)
```

```
## Variable Loadings and Descriptive Information: Dimension 1
## Variable Name Cases Loading Mean Std Dev
## Atlas 14 0.6370 28.00000 4.21307
## Datafolha 172 0.9851 33.44448 18.68127
## DataPoder360 24 0.9743 22.04167 13.43341
## Gallup 63 0.9902 32.66667 17.33608
## IBOPE 142 0.9855 36.55540 18.18718
## IBPAD 6 -0.6322 34.66667 3.27652
## IdeiaBigData 28 0.9439 33.73214 5.95891
## IPESPE 31 0.9740 29.12903 11.10287
```

```
##      IPSOS      45      0.9770  5.55556  5.00469
##      MDA       28      0.9711 25.45714 16.70788
##      Offerwise  6      0.0196 33.95000  2.17390
##      Parana     5      0.8724 35.50000  2.08135
##      Quaest     5      0.7513 24.80000  3.65513
##      Sensus    71      0.9929 41.80000 17.24610
##      Vox       45      0.9789 24.13333 13.45214
```

```
wcalc.M <- data.frame(M=gsub("\\.", "-", wcalc.Mraw$period, perl=T)
                      ,latentM=wcalc.Mraw$latent1)
wcalc.M$M <- gsub("-1$", "-10", wcalc.M$M )
```

```
### Merge WCALC, and averaging estimates:
```

```
### Raw estimates no longer saved (look at raw file, instead)
```

```
dm <- merge(ms, wcalc.M, by=c("M"), all=T)
```

```
### Fill in missing presidents names (for those months for which we had not data)
```

```
### This is based on dates, so first impute day of month for missing observations
```

```
### For mnth, take center of month, doesn't matter because never two presidents
```

```
dm$date <- as.Date(paste(dm$M, "-15", sep=""))
```

```
#Enter the dates of presidencies#####
```

```
pres.dates <- c(
```

```
  as.Date(c(
```

```
    "1979-03-15", #start of Figueiredo, prior to start of data
```

```
    "1985-03-15", #start of Sarney
```

```
    "1990-03-15", #start of Collor
```

```
    "1992-10-02", #start of Franco
```

```
    "1995-01-01", #start of FHC
```

```
    "2003-01-01", #start of Lula
```

```
    "2011-01-01", #start of Dilma
```

```
    "2016-08-31", #start of Temer
```

```
    "2019-01-01" #start of Bolsonaro
```

```
 )), Sys.Date())
```

```
dm$PresidentS <- dm$presUsed
```

```
missing.pres <- dm$PresidentS[is.na(dm$PresidentS)]
```

```
missing.dates <- dm$date[is.na(dm$PresidentS)]
```

```
dm$PresidentS[is.na(dm$PresidentS)] <- ifelse(
```

```
  missing.dates < pres.dates[2], "FIGUEIREDO", ifelse(missing.dates >= pres.dates[2] & missing.dates < pres.dates[3], "DILMA", "BOLSONARO"))
```

```
dm$PresidentS <- factor(dm$PresidentS, levels=c("FIGUEIREDO", "SARNEY", "COLLOR", "FRANCO",
                                              "CARDOSO", "LULA", "DILMA", "TEMER", "BOLSONARO"))
```

```
## This is the same for both datasets (record presidents that finished term, etc)
```

```
elected.pres <- levels(dm$PresidentS)[-c(1,2,4)]
```

```
concluded.pres <- levels(dm$PresidentS)[-c(3,7)]
```

```
### Add linear interpolations for average approach
```

```
### We do this by president so as not to interpolate at end and start
```

```
### At end and start, repeat first or last obser
```

```

allpres <- levels(dm$PresidentS)
dm$popM.li <- NA
for(pp in allpres){
  l <- min(which(is.na(dm$popM) == F & dm$PresidentS == pp))
  h <- max(which(is.na(dm$popM) == F & dm$PresidentS == pp))
  dm$popM.li[l:h] <- data.frame(dm$popM[l:h],
    approx(dm$popM[l:h], method = "linear", n = length(dm$popM[l:h])))$y

  hh <- max(which(dm$PresidentS == pp))
  ll <- min(which(dm$PresidentS == pp))
  if(hh > h){#if there is missing at the end of term, impute average of last values
    # (and project in LatentM)
    dm$popM.li[(h+1):hh] <- mean(dm$popM.li[(h-2):h])
    m.to.fill <- length((h+1):hh) ##and for LatentM->linearly project from last 3 points
    dm$latentM[(h+1):hh] <- approx(dm$latentM[(h-2):h], n = (3+m.to.fill))$y[-c(1:3)]
  }
  if(ll < l){#if there is missing at start of term, impute average of first values
    dm$popM.li[ll:(l-1)] <- mean(dm$popM.li[l:(l+1)])
    dm$latentM[ll:(l-1)] <- mean(dm$latentM[l:(l+1)])
  }
}

#Compute the counter for months in the term for each observation
dm$minterm <- round(as.numeric(dm$date-pres.dates[as.numeric(dm$PresidentS)])/m1, 1)

#Honey moon indicator, but only for elected presidents
dm$hm <- ifelse(dm$minterm <= 4 &
  is.element(dm$PresidentS, elected.pres), T, F)
dm$hmc <- ifelse(dm$minterm <= 6 &
  is.element(dm$PresidentS, elected.pres), abs(dm$minterm-6), 0)

#Compute months left in term
dm$mleft <- round(
  ifelse(as.numeric(dm$PresidentS) == max(as.numeric(dm$PresidentS)),
    NA, #last president, can't compute months left in term
    as.numeric(pres.dates[1+as.numeric(dm$PresidentS)]-dm$date)/m1), 1)

#Compute lame duck indicator
dm$ld <- ifelse(is.na(dm$mleft), F, #last president is NA
  dm$mleft <= 4 & is.element(dm$PresidentS, concluded.pres))

```

3.2. Summary statistics

```

load("R/popularity_raw_BR.RData")
load("R/popularity_raw_bolsonaro_BR.RData")

Nall <- nrow(dd)

d <- subset(d, President != "Figueiredo")
dm <- subset(dm, PresidentS != "FIGUEIREDO")
Nused <- sum(is.na(d$Positive) != T) #total number of raw observations used

```

```

Npollsters <- length(unique(d$Institute))
Nmonths <- nrow(dm) #months spanned by the monthly dataset
Nmonthsdata <- nrow(ms) #months in which there was some observation
Nimp <- sum(is.na(dm$popM))
N1 <- min(dm$M)
NN <- max(dm$M)

obs <- data.frame(Nall,Nused,Npollsters,Nmonths,Nimp
                  ,First=N1,Last=NN)

obs<-list(summary=obs,by.pollster=as.matrix(table(ds$Varname)))
save(obs,file="DATA/obs_BR.RData")
write.csv(obs,'DATA/obs_BR-M.csv')

```

3.3. Save the datasets

```

dm <- subset(dm,select=c(date,M,PresidentS,minterm,hm,hmc,ld,
                        popM,popM.li,latentM,instituteM))
dm$term <- as.character(dm$PresidentS)
dm$term[which(dm$date>as.Date("1999-01-01")&dm$PresidentS=="CARDOSO")] <- "CARDOSO II"
dm$term[which(dm$date>as.Date("2007-01-01")&dm$PresidentS=="LULA")] <- "LULA II"
dm$term[which(dm$date>as.Date("2015-01-01")&dm$PresidentS=="DILMA")] <- "DILMA II"
dm$country<-"Brazil"
save(dm,file="DATA/data_BR-M.RData")
write.csv(dm,'DATA/data_BR-M.csv')

cat("\nCorrelation between MONTHLY linear imputed and Wcalc:\n")

```

```

##
## Correlation between MONTHLY linear imputed and Wcalc:

```

```

print(cor.test(dm$popM.li,dm$latentM))

```

```

##
## Pearson's product-moment correlation
##
## data: dm$popM.li and dm$latentM
## t = 115.34, df = 434, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9808070 0.9867923
## sample estimates:
##      cor
## 0.9840762

```


4. Plotting

4.1. Merge estimates into d dataset for plotting

```
d <- merge(d,subset(dm,select=c(M,popM.li,latentM)),by="M",all=T)
d <- d[order(d$M),]#make sure data are ordered
save(d,file="DATA/data_BR-D.RData")
write.csv(d,'DATA/data_BR-D.csv')

## Save popularity at election time
elec.date <- as.Date(c("1988-11-15","1989-11-15","1990-03-10",
                      "1992-10-03","1994-10-03","1996-10-03",
                      "1998-10-04","2000-10-01","2002-10-06",
                      "2004-10-03","2006-10-01",
                      "2008-10-05","2010-10-03",
                      "2012-10-07","2014-10-05","2016-10-01",
                      "2018-10-07"))
pop.elec <- data.frame(matrix(NA,nrow=2,ncol=length(elec.date),
                             dimnames=list(c("popM.li","latentM"),
                             c(as.character(elec.date)))))
#popularity of presidents close to election

for(i in 1:length(elec.date)){
  pop.elec[1,i] <- d$popM.li[which.min(abs(as.numeric(d$date-elec.date[i])))]
  pop.elec[2,i] <- d$latentM[which.min(abs(as.numeric(d$date-elec.date[i])))]
}
pop.elec <- t(pop.elec)
save(pop.elec,file="DATA/data_BR_elections.RData")
write.csv(pop.elec,'DATA/data_BR-elections.csv')
```

4.2. Plot

```
par(mar = c(2.5,5.5,.5,.5))
min.y <- 0
max.y <- 100
plot(d$date, d$Positive,type="n",
     ylab="Approval or Popularity",
     xlab="", bty="n",
     cex.axis = 1.2, cex.lab = 1.2, ylim = c(min.y,max.y))

polygon(x = c(min(d$date), pres.dates[1], pres.dates[1], min(d$date)),
       y = c(min.y, min.y, max.y, max.y), border = NA, col = gray(0.9))

for(i in seq(2, length(pres.dates), by = 2)){
  polygon(x = c(pres.dates[i], pres.dates[i+1], pres.dates[i+1], pres.dates[i]),
        y = c(min.y, min.y, max.y, max.y), border = NA, col=gray(0.9))
}

points(d$date, d$popM, pch = ".", cex = 2)
alt <- -1
```

```

for(i in levels(d$PresidentS)){
  text(mean(d$date[d$PresidentS==i], na.rm=T), max.y-2, labels=i, cex=0.6, pos=2+alt)
  lines(d$date[which(d$PresidentS==i)], d$latentM[which(d$PresidentS==i)], col=gray(0))
  lines(d$date[which(d$PresidentS==i)], d$popM.li[which(d$PresidentS==i)], col=1, lty=3)
  alt <- alt * -1 #to alternate position of name
}
legend(x = as.Date("2002-01-01"), y = 20,
       legend = c("Raw Data Point", "Average (Monthly)", "Latent Estimate (Monthly)",
                  "Latent Estimate (Quarterly)"),
       cex = 0.8,
       lty = c(NA, 3, 1, 1),
       col = c(1, gray(0), gray(.5)),
       pch = c(".", NA, NA, NA), pt.cex = 4, bty = "n")

#abline(h = 33, lty = 2)
abline(v = c(as.Date("1994-10-01"),
             as.Date("1998-10-01"),
             as.Date("2002-10-01"),
             as.Date("2006-10-01"),
             as.Date("2010-10-01"),
             as.Date("2014-10-01"),
             as.Date("2018-10-01")))

```

