Basic population estimates with British Social Attitudes Survey data using R

UK Data Service

April 2024

This exercise is part of the [‘Introduction to the British Social Attitudes Survey (BSA)’](https://trainingmodules.ukdataservice.ac.uk/attitudes/" \l "/) online module. In the exercise, we examine data from the 2020 British Social Attitudes survey to find out:

* what proportion of respondents said they voted remain in the EU Referendum?
* whether people think the government should raise taxes and spend more or reduce tax and cut social expenditures?
* how much people think they’ll get from the State pension?

Answers to the questions asked throughout the exercise can be found at the end of the page.

### Getting started

Data can be downloaded from the [UK Data Service website](https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=9005) following [registration](https://ukdataservice.ac.uk/help/registration/registration-login-faqs/). Download the compressed folder, unzip and save it somewhere accessible on your computer.

The examples below assume that the dataset has been saved in a new folder named *UKDS* on your Desktop (Windows computers). The path would typically be C:\Users\YOUR\_USER\_NAME\Desktop\UKDS. Feel free to change it to the location that best suits your needs.

We begin by loading the R packages needed for the exercise and set the working directory.

library(dplyr) ### Data manipulation functions  
library(haven) ### Functions for importing data from   
 ### commercial packages  
library(Hmisc) ### Extra statistical functions  
  
### Setting up the working directory  
### Change the setwd() command to match the location   
### of the data on your computer   
### if required   
  
setwd("C:\Users\Your\_Username\_here\")  
  
getwd()  
  
# Opening the BSA dataset in SPSS format  
  
bsa20<-read\_spss(  
 'UKDA-9005-spss/spss/spss25/bsa2020\_archive.sav'  
 )

[1] C:\Users\Your\_Username\_here\

### 1. Explore the dataset

Start by getting an overall feel for the data. Either inspect variables and cases in the data editor or use the code below to produce a summary of all the variables in the dataset.

### Gives the number of rows (observations)   
### and columns (variables)  
dim(bsa20)

[1] 3964 210

### List variable names in their actual   
 ### order in the dataset  
names(bsa20)

[1] "serial" "QnrVersion" "RespSx2cat" "RespAgeE" "MarStat6"   
 [6] "REconFW01" "REconFW02" "REconFW03" "REconFW04" "REconFW05"   
 [11] "REconFW06" "REconFW07" "REconFW08" "REconFW09" "REconFW10"   
 [16] "REconFW11" "EMPSTAT" "Employ" "Superv" "EmpOCC"   
 [21] "TenureE" "SupParty" "ClosePty" "PARTYFW" "Idstrng"   
 [26] "RemLea" "RemLeaCl" "RemLeaSt" "Politics" "ConLabDf"   
 [31] "VoteDuty" "SocTrust" "EngParl" "ScotPar2" "ECPolicy2"   
 [36] "Spend1" "Spend2" "SocBen1" "SOCBEN2" "DOLE"   
 [41] "TAXSPEND" "WkMent" "WkPhys" "HProbRsp" "PhsRetn"   
 [46] "PhsRecov" "MntRetn" "MntRecov" "HCWork21" "HCWork22"   
 [51] "HCWork23" "HCWork24" "HCWork25" "HCWork26" "HCWork28"   
 [56] "HCWork29" "HCWork213" "HCWork214" "HCWork215" "HCWork27"   
 [61] "CMtUnmar1" "CMtUnmar2" "CMtUnmar3" "CMtUnmar4" "CMtUnmar5"   
 [66] "CMtUnmar6" "CMtUnmar7" "CMtUnmar8" "CMtUnmar9" "CMtUnmar10"   
 [71] "CMtmar1" "CMtmar2" "CMtmar3" "CMtmar4" "CMtmar5"   
 [76] "CMtmar6" "CMtmar7" "CMtmar8" "CMtmar9" "CMtmar10"   
 [81] "ChCoSupp" "ChMIncM" "ChMIncF" "ChMCont" "RBGaran2"   
 [86] "RBGGov" "DigPCUn" "DigPCctl" "DigPCcon" "DigPCrsk"   
 [91] "DigGVun" "DigGVctl" "DigGVcon" "DigGVrsk" "DigPro"   
 [96] "NHSSat" "WkHmNow" "WkHmJan" "CovWkc" "CovNoWkc"   
[101] "CovWkr1" "CovWkr2" "CovWkr3" "CovWkr4" "CovWkr5"   
[106] "CovWkr6" "CovWk1" "CovWk2" "CovWk3" "GovtWork"   
[111] "GovTrust" "CLRTRUST" "MPsTrust" "LoseTch" "VoteIntr"   
[116] "PtyNMat2" "PolPart01" "PolPart02" "PolPart03" "PolPart04"   
[121] "PolPart05" "PolPart06" "PolPart07" "PolPart08" "PolPart09"   
[126] "PolPart10" "PolPart11" "REFHANG" "RefSyst" "UnempJob"   
[131] "SocHelp" "DoleFidl" "WelfFeet" "welfhelp" "morewelf"   
[136] "damlives" "proudwlf" "Redistrb" "BigBusnN" "Wealth"   
[141] "RichLaw" "Indust4" "TradVals" "StifSent" "DeathApp"   
[146] "Obey" "WrongLaw" "Censor" "NatIdGB" "ChAttend"   
[151] "DisNew2" "DisAct" "HEdQual2" "HhldEdu" "EURefV2"   
[156] "EUVOTWHO" "EURefb" "Voted" "Vote" "Anybn3"   
[161] "HHincome" "Maininc5" "REarn" "HIncDif4" "RetExp"   
[166] "RetExpb" "FutrWrk" "PenKnow2" "PenExp2" "PenComp"   
[171] "PenIntr" "INFORET3" "WkPKnw" "WKPSav" "WkPSpn"   
[176] "WPSvUs" "WPSvWw" "WPSvEas" "PrPKnw" "PrPSav"   
[181] "PrPSpn" "PrPSvUs" "PrPSvWW" "PrPSvEas" "NCOutcome"   
[186] "Ragecat" "Ragecat20" "DisActDV" "leftrigh" "libauth"   
[191] "welfare2" "libauth2" "leftrig2" "welfgrp" "REconAct20"   
[196] "REconSum20" "RaceOri4" "LegMarStE" "HhlAdGpd" "HhlChlGpd"   
[201] "BestNatU2" "RetirAg3" "ReligSum20" "RlFamSum20" "EmplStatDV"   
[206] "RClassGP" "serialh" "GOR" "gor2" "BSA20\_wt\_new"

### Displays the first five  
### lines of a data frame  
### Beware, the output might be lengthy!  
head(data.frame(bsa20))

serial QnrVersion RespSx2cat RespAgeE MarStat6 REconFW01 REconFW02  
1 3.211e+09 1 2 70 5 0 0  
2 3.211e+09 1 2 66 1 0 0  
3 3.211e+09 1 1 64 1 0 0  
4 3.211e+09 1 2 43 1 0 0  
5 3.211e+09 1 1 38 1 0 0  
6 3.211e+09 1 2 77 1 0 0  
 REconFW03 REconFW04 REconFW05 REconFW06 REconFW07 REconFW08 REconFW09  
1 0 0 0 0 0 0 1  
2 0 0 0 0 0 0 1  
3 0 0 0 0 0 0 1  
4 1 0 0 0 0 0 0  
5 1 0 0 0 0 0 0  
6 0 0 0 0 0 0 1  
 REconFW10 REconFW11 EMPSTAT Employ Superv EmpOCC TenureE SupParty ClosePty  
1 0 0 1 2 1 3 10 1 NA  
2 0 0 1 2 1 1 1 1 NA  
3 0 0 1 1 2 1 1 1 NA  
4 0 0 1 3 1 3 1 2 2  
5 0 0 1 3 2 2 1 2 2  
6 0 0 3 NA NA 1 9 1 NA  
 PARTYFW Idstrng RemLea RemLeaCl RemLeaSt Politics ConLabDf VoteDuty SocTrust  
1 1 2 NA NA NA 2 NA NA 1  
2 2 3 NA NA NA 3 NA NA 1  
3 2 3 NA NA NA 3 NA NA 1  
4 2 3 NA NA NA 2 NA NA 2  
5 1 3 NA NA NA 3 NA NA 2  
6 1 2 NA NA NA 2 NA NA 2  
 EngParl ScotPar2 ECPolicy2 Spend1 Spend2 SocBen1 SOCBEN2 DOLE TAXSPEND WkMent  
1 NA NA NA 2 1 1 2 1 2 1  
2 NA NA NA 1 3 2 5 1 2 2  
3 NA NA NA 3 1 2 3 1 2 2  
4 NA NA NA 7 3 1 2 2 2 2  
5 NA NA NA 7 3 2 4 2 2 1  
6 NA NA NA 98 NA 1 4 2 3 2  
 WkPhys HProbRsp PhsRetn PhsRecov MntRetn MntRecov HCWork21 HCWork22 HCWork23  
1 1 1 1 2 1 2 1 1 1  
2 2 1 1 3 1 2 1 0 1  
3 2 1 1 2 1 2 1 1 1  
4 2 2 2 3 1 2 1 1 1  
5 1 1 1 2 1 2 1 1 1  
6 2 2 2 2 2 2 1 0 1  
 HCWork24 HCWork25 HCWork26 HCWork28 HCWork29 HCWork213 HCWork214 HCWork215  
1 1 1 1 0 0 0 0 0  
2 1 1 1 0 0 0 0 0  
3 1 1 1 0 0 0 0 0  
4 1 1 1 0 0 0 0 0  
5 1 1 1 0 0 0 0 0  
6 1 1 0 0 0 0 0 0  
 HCWork27 CMtUnmar1 CMtUnmar2 CMtUnmar3 CMtUnmar4 CMtUnmar5 CMtUnmar6  
1 0 1 2 2 1 1 1  
2 0 1 1 1 3 3 1  
3 0 1 1 1 3 3 1  
4 0 NA NA NA NA NA NA  
5 0 NA NA NA NA NA NA  
6 0 1 1 1 3 1 8  
 CMtUnmar7 CMtUnmar8 CMtUnmar9 CMtUnmar10 CMtmar1 CMtmar2 CMtmar3 CMtmar4  
1 1 2 1 1 NA NA NA NA  
2 1 1 3 1 NA NA NA NA  
3 1 1 3 3 NA NA NA NA  
4 NA NA NA NA 1 1 2 1  
5 NA NA NA NA 1 1 1 1  
6 1 1 3 1 NA NA NA NA  
 CMtmar5 CMtmar6 CMtmar7 CMtmar8 CMtmar9 CMtmar10 ChCoSupp ChMIncM ChMIncF  
1 NA NA NA NA NA NA 3 1 NA  
2 NA NA NA NA NA NA 3 2 NA  
3 NA NA NA NA NA NA 2 2 NA  
4 1 1 1 2 1 1 NA NA 1  
5 1 1 1 2 1 1 NA NA 1  
6 NA NA NA NA NA NA 3 8 NA  
 ChMCont RBGaran2 RBGGov DigPCUn DigPCctl DigPCcon DigPCrsk DigGVun DigGVctl  
1 1 2 NA 2 2 2 1 NA NA  
2 4 2 NA 2 3 3 1 NA NA  
3 2 3 NA 3 3 3 8 NA NA  
4 NA NA NA NA NA NA NA 1 2  
5 NA NA NA NA NA NA NA 3 3  
6 1 1 1 1 3 1 2 NA NA  
 DigGVcon DigGVrsk DigPro NHSSat WkHmNow WkHmJan CovWkc CovNoWkc CovWkr1  
1 NA NA 2 3 NA NA NA NA NA  
2 NA NA 2 2 NA NA NA NA NA  
3 NA NA 2 3 NA NA NA NA NA  
4 4 1 2 2 1 2 NA 1 0  
5 3 8 1 2 3 3 1 NA 0  
6 NA NA 2 2 NA NA NA NA NA  
 CovWkr2 CovWkr3 CovWkr4 CovWkr5 CovWkr6 CovWk1 CovWk2 CovWk3 GovtWork  
1 NA NA NA NA NA NA NA NA NA  
2 NA NA NA NA NA NA NA NA NA  
3 NA NA NA NA NA NA NA NA NA  
4 0 0 0 1 0 5 5 5 NA  
5 0 0 0 0 1 3 3 3 NA  
6 NA NA NA NA NA NA NA NA NA  
 GovTrust CLRTRUST MPsTrust LoseTch VoteIntr PtyNMat2 PolPart01 PolPart02  
1 NA NA NA NA NA NA NA NA  
2 NA NA NA NA NA NA NA NA  
3 NA NA NA NA NA NA NA NA  
4 NA NA NA NA NA NA NA NA  
5 NA NA NA NA NA NA NA NA  
6 NA NA NA NA NA NA NA NA  
 PolPart03 PolPart04 PolPart05 PolPart06 PolPart07 PolPart08 PolPart09  
1 NA NA NA NA NA NA NA  
2 NA NA NA NA NA NA NA  
3 NA NA NA NA NA NA NA  
4 NA NA NA NA NA NA NA  
5 NA NA NA NA NA NA NA  
6 NA NA NA NA NA NA NA  
 PolPart10 PolPart11 REFHANG RefSyst UnempJob SocHelp DoleFidl WelfFeet  
1 NA NA NA NA 3 4 4 4  
2 NA NA NA NA 3 3 3 4  
3 NA NA NA NA 3 4 4 4  
4 NA NA NA NA 2 3 3 1  
5 NA NA NA NA 2 4 2 3  
6 NA NA NA NA 2 2 2 2  
 welfhelp morewelf damlives proudwlf Redistrb BigBusnN Wealth RichLaw Indust4  
1 4 2 2 1 3 4 3 5 4  
2 4 3 1 2 4 3 3 4 4  
3 3 3 1 1 3 3 2 3 3  
4 2 4 3 3 4 2 2 2 3  
5 3 3 3 2 4 2 3 3 4  
6 3 3 4 2 4 4 3 5 4  
 TradVals StifSent DeathApp Obey WrongLaw Censor NatIdGB ChAttend DisNew2  
1 3 3 2 3 4 3 5 7 2  
2 4 3 2 2 3 2 6 NA 2  
3 3 3 3 2 2 2 1 NA 2  
4 2 1 2 1 2 2 3 7 2  
5 4 3 3 3 4 2 3 NA 2  
6 1 2 3 1 3 2 3 1 2  
 DisAct HEdQual2 HhldEdu EURefV2 EUVOTWHO EURefb Voted Vote Anybn3 HHincome  
1 NA 2 2 NA NA NA 2 NA 1 2  
2 NA 1 NA NA NA NA 1 2 2 3  
3 NA 2 1 NA NA NA 1 2 2 3  
4 NA 4 2 NA NA NA 1 1 1 4  
5 NA 3 2 NA NA NA 1 1 1 3  
6 NA 1 NA NA NA NA 1 1 1 9  
 Maininc5 REarn HIncDif4 RetExp RetExpb FutrWrk PenKnow2 PenExp2 PenComp  
1 4 NA 3 NA NA NA NA NA NA  
2 2 NA 2 NA NA NA NA NA NA  
3 2 NA 2 NA NA NA NA NA NA  
4 1 3 2 3 60 2 1 7000 4  
5 1 3 3 3 65 1 2 130 2  
6 1 NA 3 NA NA NA NA NA NA  
 PenIntr INFORET3 WkPKnw WKPSav WkPSpn WPSvUs WPSvWw WPSvEas PrPKnw PrPSav  
1 NA NA NA NA NA NA NA NA NA NA  
2 NA NA NA NA NA NA NA NA NA NA  
3 NA NA NA NA NA NA NA NA NA NA  
4 2 2 2 1 4 1 1 1 NA NA  
5 2 2 3 1 4 1 2 2 NA NA  
6 NA NA NA NA NA NA NA NA NA NA  
 PrPSpn PrPSvUs PrPSvWW PrPSvEas NCOutcome Ragecat Ragecat20 DisActDV leftrigh  
1 NA NA NA NA 1 7 6 3 3.8  
2 NA NA NA NA 1 7 6 3 3.6  
3 NA NA NA NA 1 6 5 3 2.8  
4 NA NA NA NA 1 3 3 3 2.6  
5 NA NA NA NA 1 3 3 3 3.2  
6 NA NA NA NA 1 7 7 3 4.0  
 libauth welfare2 libauth2 leftrig2 welfgrp REconAct20 REconSum20 RaceOri4  
1 3.000000 2.000 2 3 1 9 6 3  
2 3.333333 2.375 2 3 1 9 6 3  
3 3.500000 2.125 2 2 1 9 6 3  
4 4.333333 3.625 3 2 3 3 2 3  
5 2.833333 3.000 2 2 2 3 2 3  
6 4.000000 3.500 3 3 2 9 6 3  
 LegMarStE HhlAdGpd HhlChlGpd BestNatU2 RetirAg3 ReligSum20 RlFamSum20  
1 4 1 0 1 65 3 1  
2 1 2 0 3 58 5 2  
3 1 2 0 1 54 5 1  
4 1 2 1 1 NA 3 2  
5 1 2 1 2 NA 5 3  
6 1 2 0 2 99 3 3  
 EmplStatDV RClassGP serialh GOR gor2 BSA20\_wt\_new  
1 4 1 321100002 1 1 0.7099859  
2 6 1 321100014 1 1 0.3145871  
3 7 1 321100014 1 1 0.5649618  
4 4 1 321100040 1 1 0.9355446  
5 7 2 321100040 1 1 0.6830794  
6 3 1 321100042 1 1 1.4006989

**Questions**

1. What is the overall sample size?
2. How many variables are there in the dataset?

Now, focus on the three variables we will use.

**Note** In traditional statistical software packages such as SPSS or Stata, categorical variables are coded as arbitrary numbers, to which values labels are attached that describe the substantive meaning of these values. R on the other hand can either directly deal with the value themselves as alphanumeric variables, or with its own version of categorical variables, known as ‘factors’. There aren’t straightforward ways to convert SPSS or Stata labelled categorical variables into R factors. The approach followed by the Haven package that we use here consist in preserving the original numeric values in the data, and add attributes that can be manipulated separately. Attributes are a special type of R objects that have a name, and can be read using the attr() function. Each variable has a ‘label’ and ‘labels’ attribute. The former is the variable description, the latter the value labels. Alternatively, haven-imported numeric variables can be converted into factors with levels (ie categories) reflecting the SPSS or Stata value labels, but with numeric values different from the original ones.

Let’s examine the original variable description and value labels.

# We can do this variable by variable...  
attr(bsa20$TAXSPEND,"label")

[1] "If it had to choose, should govt reduce/increase/maintain levels of taxation and spending?"

# Or all as once  
t(bsa20|>select(TAXSPEND,EUVOTWHO,PenExp2)  
 |>summarise\_all(attr,"label"))

[,1]   
TAXSPEND "If it had to choose, should govt reduce/increase/maintain levels of taxation and spending?"   
EUVOTWHO "Did you vote to 'remain a member of the EU' or to 'leave the EU'?"   
PenExp2 "How much do you think someone who reaches State Pension age today would receive in pounds per week?"

# The same holds with value labels  
attr(bsa20$TAXSPEND,"labels")

Not applicable   
 -1   
 Reduce taxes and spend less on health, education and social benefits   
 1   
 Keep taxes and spending on these services at the same level as now   
 2   
Increase taxes and spend more on health, education and social benefits   
 3   
 Don't know   
 8   
 Prefer not to answer   
 9

attr(bsa20$EUVOTWHO,"labels")

Not applicable Remain a member of the European Union   
 -1 1   
 Leave the European Union I Don't remember   
 2 3   
 Don't know Prefer not to answer   
 8 9

**Question 3** What do the variables measure and how?

### 2. Missing values

Let’s now examine the distribution of our three variables. We can temporarily convert EUVOTWHO and TAXSPEND into factors using mutate() for a more meaningful output that include their value labels. Review the frequency tables, examining the ‘not applicable’ and ‘don’t know’ categories.

bsa20%>%select(EUVOTWHO,TAXSPEND) %>%  
 mutate(as\_factor(.)) %>%  
 summary()

EUVOTWHO   
 Not applicable : 0   
 Remain a member of the European Union: 635   
 Leave the European Union : 463   
 I Don't remember : 2   
 Don't know : 0   
 Prefer not to answer : 21   
 NA's :2843   
 TAXSPEND   
 Not applicable : 0   
 Reduce taxes and spend less on health, education and social benefits : 186   
 Keep taxes and spending on these services at the same level as now :1589   
 Increase taxes and spend more on health, education and social benefits:2133   
 Don't know : 35   
 Prefer not to answer : 21

summary(bsa20$PenExp2)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
 0 120 160 1293 200 9999 1076

**Question 4** Why for EUVOTWHO and PenExp2 are there so many system missing values (NA)? Note, you can use the documentation to check if needed. What does this mean when it comes to interpreting the percentages?

When analysing survey data, it is sometimes convenient to set all item non responses such as ´Don’t know´ and ‘Prefer not to say’ as system missing so that they do not appear in the results. With BSA data, this can only be done as long as we do not weight the data (in order to make inference to the British population as a whole), otherwise we might get biased results.

The code below show how to recode the missing values into system missing (NA) using separate variables. For ease of interpretation, we also convert the original numeric variable into labelled factors using as\_factor(), so that they directly display the value labels.

bsa20<-bsa20%>%mutate(  
 TAXSPEND.r=factor(as\_factor(TAXSPEND,"labels"),   
 exclude = c("Prefer not to answer",  
 "Don't know")),  
 EUVOTWHO.r=factor(as\_factor(EUVOTWHO,"labels"),  
 exclude = c("Prefer not to answer",  
 "I Don't remember","Not applicable",NA)),  
 PenExp2.r=ifelse(PenExp2==-1 | PenExp2>=9998,NA,PenExp2)  
 )  
### Value labels need to be truncated as they are rather lengthy!  
levels(bsa20$TAXSPEND.r)<-substr(levels(bsa20$TAXSPEND.r),1,14)  
levels(bsa20$EUVOTWHO.r)<-substr(levels(bsa20$EUVOTWHO.r),1,6)  
  
levels(bsa20$TAXSPEND.r)

[1] "Reduce taxes a" "Keep taxes and" "Increase taxes"

levels(bsa20$EUVOTWHO.r)

[1] "Remain" "Leave "

### 3. Compare unweighted and weighted proportions

Let’s examine the unweighted responses first. In order to ensure coherence with the remainder of this exercise, we use xtabs() for categorical variables and summary() for continuous ones.

Unlike some other surveys, the BSA has retained observations with “Don’t knows” and ‘Does not apply’ when weights were computed. As a result, any univariate analysis needs to retain these observations, otherwise the estimated results might be incorrect. There is a way of obtaining ‘clean’ estimates without them, but this goes beyond this exercise. Instead, we will examine below results that includes them.

bsa20<-bsa20%>%mutate(  
 TAXSPEND.f=as\_factor(TAXSPEND,"labels"),   
 EUVOTWHO.f=as\_factor(EUVOTWHO,"labels"),  
 PenExp2.=ifelse(PenExp2==-1 | PenExp2>=9998,NA,PenExp2)  
 )  
  
# As before, we can truncate factor levels for a more human-friendly output  
levels(bsa20$TAXSPEND.f)<-substr(levels(bsa20$TAXSPEND.f),1,14)  
levels(bsa20$EUVOTWHO.f)<-substr(levels(bsa20$EUVOTWHO.f),1,6)  
  
  
  
  
round( ### Rounds the results to one decimal  
 100\* ### Converts proportions to %   
 prop.table( ### Computes proportions  
 xtabs(~TAXSPEND.f,bsa20, ### Computes frequencies,  
 drop.unused.levels = T) ### Leaves out levels with 0 observations),  
 ),   
 1)

TAXSPEND.f  
Reduce taxes a Keep taxes and Increase taxes Don't know Prefer not to   
 4.7 40.1 53.8 0.9 0.5

round(100\*prop.table(xtabs(~EUVOTWHO.f,bsa20,drop.unused.levels = T)),1)

EUVOTWHO.f  
Remain Leave I Don' Prefer   
 56.6 41.3 0.2 1.9

summary(bsa20$PenExp2)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
 0 120 160 1293 200 9999 1076

What is the (unweighted) percentage of respondents who say they voted remain in the EU referendum? About 58 percent of sample members who voted in referendum said they voted to remain. This figure seems a bit high (though people do not always report accurately).

Let’s compare with the weighted frequencies. We will use the wtd.table() from the Hmisc package. The weights are specified after the variable for which we request the frequencies in the command below.

# Raw output  
wtd.table(bsa20$EUVOTWHO.f,weights=bsa20$BSA20\_wt\_new)

$x  
[1] "Remain" "Leave" "I Don'" "Prefer"  
  
$sum.of.weights  
[1] 565.011079 489.146642 3.752765 22.527320

# Converted into proportions   
euv.w<-wtd.table(bsa20$EUVOTWHO.f,weights=bsa20$BSA20\_wt\_new)  
  
# Raw results  
euv.wp<-round(  
 100\*  
 prop.table(  
 euv.w$sum.of.weights),  
 1)  
  
euv.wp

[1] 52.3 45.3 0.3 2.1

# We can easily improve the output  
cbind(euv.w$x,"Weighted %"=euv.wp)

Weighted %  
[1,] "Remain" "52.3"   
[2,] "Leave" "45.3"   
[3,] "I Don'" "0.3"   
[4,] "Prefer" "2.1"

Now, what proportion say they voted remain in the EU referendum? It is about 52 percent, lower than the unweighted proportion and closer to the actual referendum results. Do you have an idea as to why this might be the case?

### 4. Confidence intervals

So far, we have just computed point estimates without worrying about their precision. Estimates precision (or uncertainty) does matter insofar as it determines how big the ranges within which ‘true’ population values are likely to be. These are also known as the *confidence intervals* of our estimates.

In this exercise, we will compute confidence intervals ‘by hand’. By doing so, we are choosing to ignore the survey design, which amounts to assuming that the sample was collected using simple random sampling - which wasn’t the case - and therefore increase the bias of our estimates. We will leave exploration of the more reliable survey design functions provided by the survey package to the next exercise.

The Hmisc package provides binconf() a handy function to compute confidence intervals for proportions. Although this is usually left under the hood by traditional statistical packages, estimating confidence intervals for proportions of categorical variables necessitates looking at each one of them individually. In other words, we need to compute one set of confidence interval for each one of the categories of TAXSPEND.f and EUVOTWHO.f separately. We can then gather them into a table of results using rbind().

We need to provide binconf() with two parameters: the frequencies for which we would like a confidence interval, and the total number of non missing observations.

### Raw confidence interval for EUVOTWHO, unweighted  
binconf(table(bsa20$EUVOTWHO.f=="Remain"),sum(!is.na(bsa20$EUVOTWHO.r)))

PointEst Lower Upper  
 0.4426230 0.4134945 0.4721515  
 0.5783242 0.5488915 0.6072108

### We can convert the output into rounded percentages for better readability.  
round(100\*  
 binconf(table(bsa20$EUVOTWHO.f=="Remain"),sum(!is.na(bsa20$EUVOTWHO.f)))[1,],  
 1)

PointEst Lower Upper   
 43.4 40.5 46.3

We can adapt the syntax above to make it work with weighted frequencies:

round(100\*  
 binconf(wtd.table(bsa20$EUVOTWHO.f,weights=bsa20$BSA20\_wt\_new)$sum.of.weights[2],  
 sum(wtd.table(bsa20$EUVOTWHO.f,weights=bsa20$BSA20\_wt\_new)$sum.of.weights)),  
 1)

PointEst Lower Upper  
 45.3 42.3 48.3

What are the differences between weighted and unweighted confidence intervals for the proportion of people who voted remain?

Let us now do the same with people’s views about government tax and spending.

w.n<-sum(wtd.table(bsa20$TAXSPEND.f,weights=bsa20$BSA20\_wt\_new)$sum.of.weights)  
  
ciprop<-cbind(levels(bsa20$TAXSPEND.f),  
round(100\*  
binconf(wtd.table(bsa20$TAXSPEND.f,weights=bsa20$BSA20\_wt\_new)$sum.of.weights,w.n),  
1)  
)  
  
ciprop

PointEst Lower Upper   
 "Not applicable" "5.5" "4.8" "6.3"   
 "Reduce taxes a" "42.8" "41.3" "44.3"  
 "Keep taxes and" "50.3" "48.8" "51.9"  
 "Increase taxes" "0.9" "0.6" "1.2"   
 "Don't know" "0.5" "0.3" "0.8"

When computing confidence intervals for means, two steps are usually needed, whether embedded in a single line of code or not: compute the mean (or any other estimate), then the confidence interval itself using confint. We also use the round() function in order to remove unneeded decimal values.

**Question 5.** What proportion think government should increase taxes and spend more on health, education and social benefits?

Several R packages offer functions for computing confidence intervals and standard errors of means. Here again, we privilege doing things by hand in order to properly undertstand what is happening in the background.

Under assumptions of simple random sampling, a 95% confidence of the mean is defined as plus or minus 1.96 times the standard error of the mean. The standard error of the mean itself is the standard error of the mean (that is, the square root of its variance) divided by the square of the sample size. Since we have functions for computing weighted means and variance in R, we can compute:

m.p<-wtd.mean(bsa20$PenExp2.r,weights=bsa20$BSA20\_wt\_new)  
se.p<-sqrt(wtd.var(bsa20$PenExp2.r,weights=bsa20$BSA20\_wt\_new))  
n<-sum(bsa20$BSA20\_wt\_new[!is.na(bsa20$PenExp2)])  
  
ci<-c(m.p,m.p-1.96\*(se.p/sqrt(n)),m.p+1.96\*(se.p/sqrt(n)))  
  
round(ci,1)

[1] 177.4 170.5 184.4

**Question 6** How much do people think they will get at state pension age?

### Answers

1. There are 3964 cases in the dataset.
2. The total number of variables is 216.
3. *TAXSPEND records responses to the questions of whether government should reduce/increase/maintain levels of taxation and spending. There are three possible responses to the question.* EUVOTWHO records responses to the question ‘Did you vote to ’remain a member of the EU’ or to ‘leave the EU’?’ The responses are ‘Remain’ or ‘Leave’. \*PenExp2 contains responses to the question ‘How much do you think someone who reaches State Pension age today would receive in pounds per week?’ Responses are numeric.
4. There are two reasons for the many not applicable.

* Routing: the question is only asked to those who said yes to a previous question (EURefV2).
* Versions 5 and 6 - The BSA uses a split sample and the question is only asked in Versions 5 and 6.

1. Between 48.8 and 51.9% in the population say the government should increase taxes and spend more.
2. The amount people think they will get at state pension age varies between £170 and £184, with an average (ie mean) in the region of £177.