# EU-voter-preferences

Joseph Arber

21/06/2020

## Table of Contents

# What determines support for the European Union?

## Introduction

In the aftermath of the UK public's vote to leave the EU in the 2016 referendum, much attention has been paid to whether support for the EU varies predictably across different types of individuals. In this question, you will use an appropriate binary dependent variable model to improve our understanding of which types of citizens are more or less likely to vote to leave the European Union if a referendum on membership were to be held in their country.

The data for this question comes from the 2016 European Social Survey (ESS) and includes information on the political attitudes and demographics of European citizens.

The question given to survey particpants was: "Imagine there were a referendum in your country tomorrow about membership of the European Union. Would you vote for your country to remain a member of the European Union or to leave the European Union?"

```
#Important packages for analysis and modelling
library(tidyverse)

## ── Attaching packages ─────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.1     ✓ dplyr   1.0.0
## ✓ tidyr   1.1.0     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0

## ── Conflicts ────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidyr)
library(dplyr)
library(texreg)

## Version:  1.37.5
## Date:     2020-06-17
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive
functions.
## Please cite the JSS article in your publications -- see
citation("texreg").

##
## Attaching package: 'texreg'
```

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

```
library(foreign)
library(ggplot2)
#library(glmnet)
#library(hrbrthemes)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

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

```
options(knitr.table.format = "html")
library(patchwork)
```

### Loading the Data

Above we have loaded the required packages for this analysis. We can now load in the data:

```
library(readr)
ess <- read_csv("data/ess.csv")
```

### Data Manipulation

We need to complete some data wrangling. There are several variables that should be converted into factor variables, this will aid the regression modelling later on, but will also provide clearer labels for the categories within the features.

```
#Variable coercion
str(ess$trade_union)
```

```
##  num [1:13075] 1 0 1 1 1 0 1 0 1 0 ...
```

```
table(ess$trade_union)
```

```
##
##    0    1
## 7910 5165
```

```
#Turn trade union into a factor variable
ess$trade_union <-factor(ess$trade_union, levels = c(0,1), labels = c("Non-
Member", "Member"))
summary(ess$trade_union)
```

```
## Non-Member     Member
##       7910       5165
```

```
class(ess$trade_union)
```

```
## [1] "factor"

#Variable coercion
str(ess$unemployed)

##  logi [1:13075] FALSE TRUE FALSE FALSE FALSE TRUE ...

table(ess$unemployed)

##
## FALSE  TRUE
## 12575   500

#Turn unemployed into a factory variable
ess$unemployed <-factor(ess$unemployed, levels = c(FALSE,TRUE), labels
=c("Employed", "Unemployed"))
summary(ess$unemployed)

##   Employed Unemployed
##      12575        500

class(ess$unemployed)

## [1] "factor"
```

There are a total of **12557** respondents who are employed, whilst there are only **500** who are unemployed. On the other hand there are around 50000 trade union members compared to 80000 non-members. We should remember these insights for the following analysis.

```
#Take a look at the level of country attachment
str(ess$country_attach)

##  num [1:13075] 8 8 9 8 8 10 4 7 9 9 ...

summary(ess$country_attach)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   7.000   8.000   8.105  10.000  10.000

#Let's sequence the country attachment variable
attach_country<-seq(0,10, length.out = 100)
summary(ess$country_attach)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   7.000   8.000   8.105  10.000  10.000

str(ess$country_attach)

##  num [1:13075] 8 8 9 8 8 10 4 7 9 9 ...
```

```r
#Take a look at the leave variable
str(ess$leave)

##  num [1:13075] 0 0 0 0 0 0 1 0 0 0 ...

table(ess$leave)

## 
##     0     1 
## 10767  2308

#Coerce leave into a factor variable
ess$leave <- factor(ess$leave, levels =c(0,1), labels = c("no","yes"))
summary(ess$leave)

##    no   yes 
## 10767  2308

table(ess$leave)

## 
##    no   yes 
## 10767  2308

str(ess$leave)

##  Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 2 1 1 1 ...
```

We used the 'seq' generator function in R, it is useful for creating proportional sequences with a given length. The rationale for doing this is that we will be able to draw more insighful conclusions by spreading the variable over a length of 100 rather than 10. The package is referenced here:

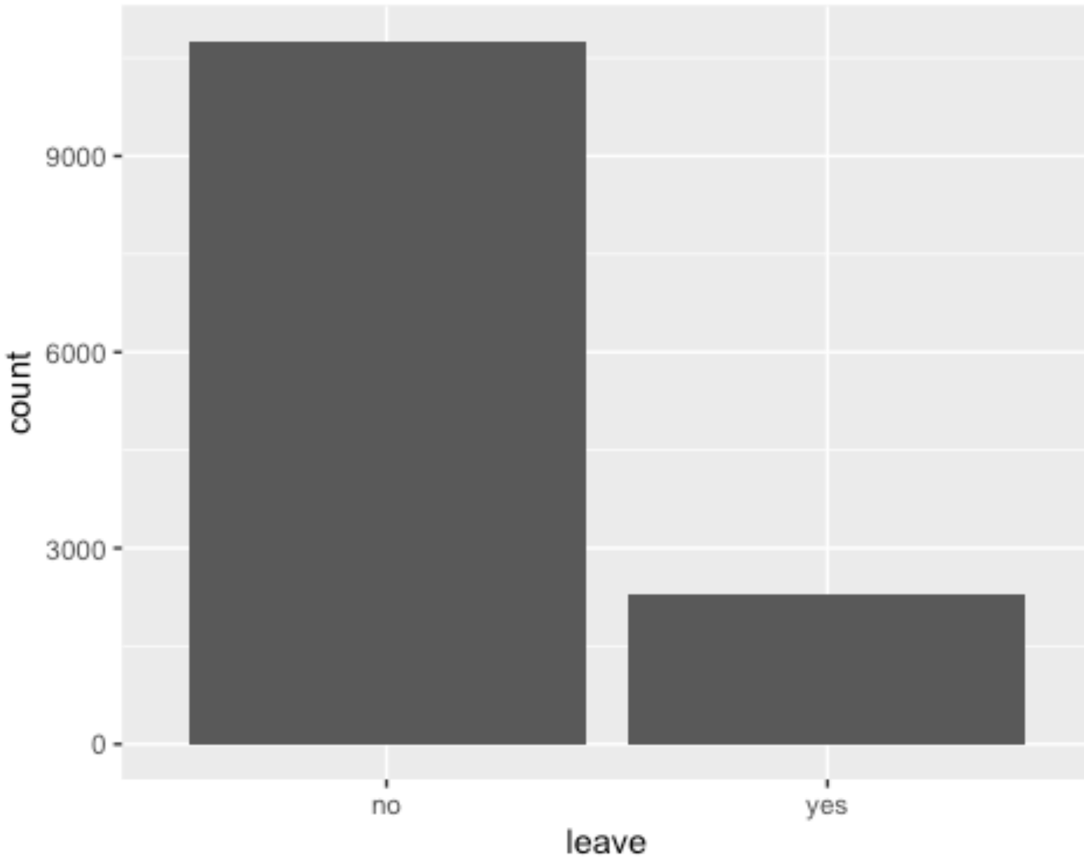
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/seq

By coercing the leave variable into a factor variable we can see that amount of people that would vote to leave was around **2308** whilst **10767** would vote to remain.

## Exploratory Analysis & Visualization

What is the vote split in the dataset?

```r
library(ggplot2)
ggplot(data = ess, aes(x = leave)) +
    geom_bar()
```




Let's try to segment the main demographic groups in the dataset. This should help us in the later modelling phases.

### *Religion and the European Union*

We will look at the categories in the religion feature.

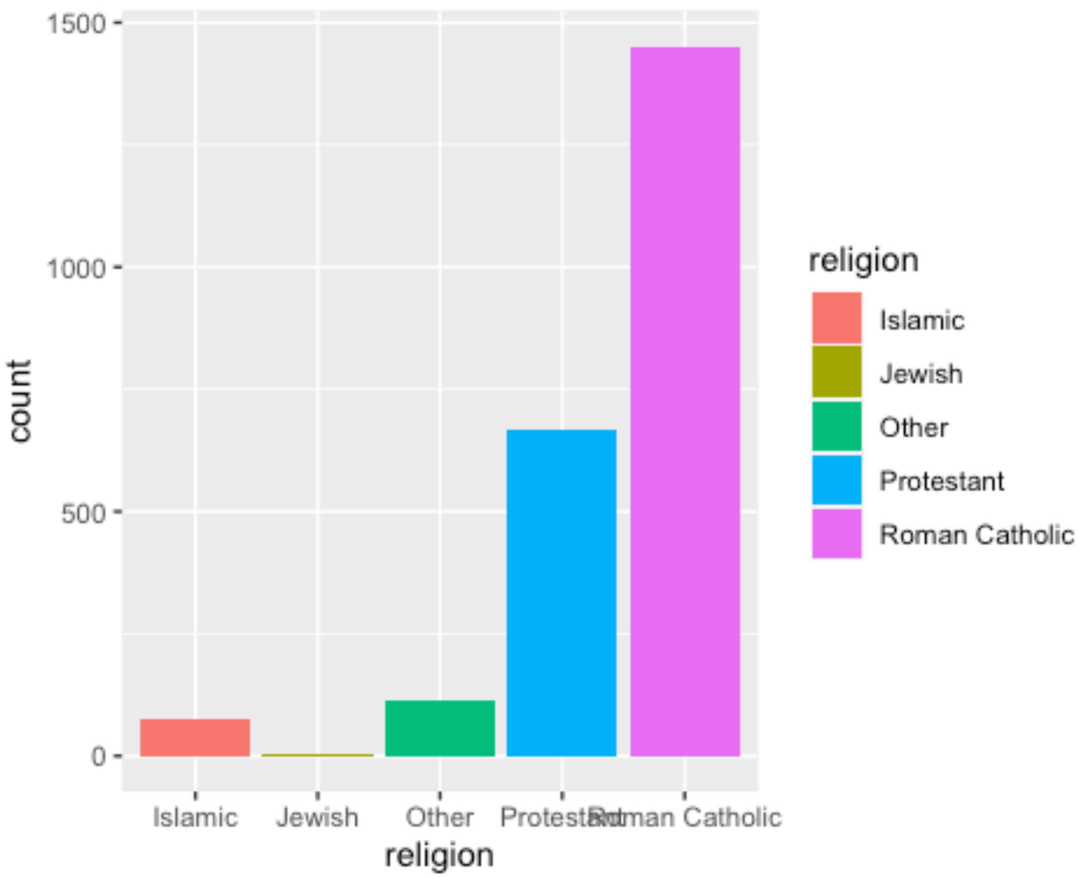
**Religious segmentations**

```
table(ess$religion)

##
##        Islamic          Jewish           Other     Protestant Roman Catholic
##            441              19             579           2826           9210
```

Including an 'Other' category, there are 5 major religions. Lets now examine which religious groups across Europe are more opposed to the EU as an institution.

```
leave_vote <- ess %>%
  filter(leave == "yes")

leave_vote

## # A tibble: 2,308 x 20
##    leave country_code gender   age years_education news_consumption
```

```
trust_people
##    <fct> <chr>        <chr>  <dbl>           <dbl>            <dbl>
<dbl>
##  1 yes   AT           Male      50              12               15
6
##  2 yes   AT           Male      45              12               35
9
##  3 yes   AT           Female    55              13               90
6
##  4 yes   AT           Male      18              12               10
6
##  5 yes   AT           Female    59              11               30
7
##  6 yes   AT           Female    23              13               30
5
##  7 yes   AT           Male      38              11              360
10
##  8 yes   AT           Female    21              12                0
10
##  9 yes   AT           Female    49              15              120
2
## 10 yes   AT           Female    58               8               30
3
## # … with 2,298 more rows, and 13 more variables: trust_politicians <dbl>,
## #   past_vote <chr>, immig_econ <dbl>, immig_culture <dbl>,
## #   country_attach <dbl>, religion <chr>, climate_change <dbl>,
## #   imp_tradition <dbl>, imp_equality <dbl>, income <dbl>,
## #   eu_integration <dbl>, trade_union <fct>, unemployed <fct>

#plot to see which religions are more opposed to the EU.
library(ggplot2)
ggplot(data = leave_vote, aes(x = religion, fill = religion)) +
    geom_bar()
```



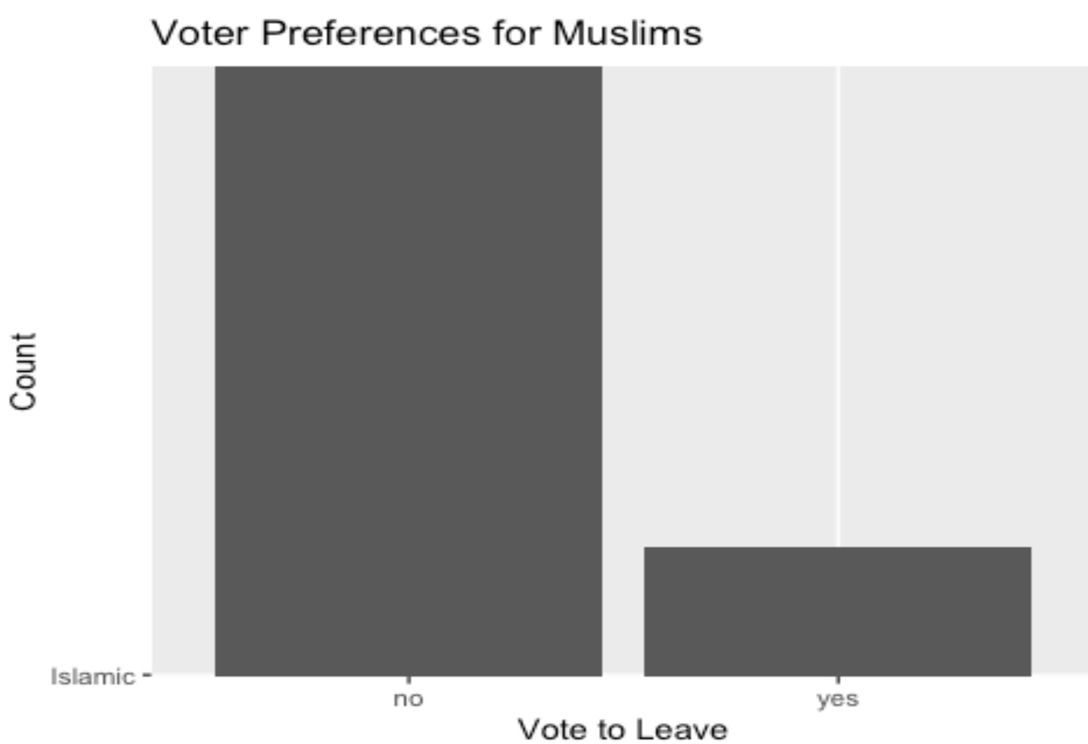

Generally **Muslims** and **Jews** are more supportive of the EU, whilst **Roman Catholics** are more opposed. Let's look at voting preferences within a religious segmentation. To do this we have to create a bucket that contains the values for all Muslims who particpated in the survey. We use the pipe operator to do this.

```
Islamic_view <- ess %>%
  filter(religion == "Islamic")

Islamic_view

## # A tibble: 441 x 20
##    leave country_code gender   age years_education news_consumption
trust_people
##    <fct> <chr>        <chr>  <dbl>           <dbl>            <dbl>
<dbl>
##  1 no    AT           Female    20              13               30
5
##  2 no    AT           Male      35              18               60
3
##  3 no    AT           Female    25              12               15
9
##  4 no    AT           Male      63               8               90
2
```

```
##   5 no    AT          Male      34              13                 0
10
##   6 no    AT          Female    18              10                30
6
##   7 no    AT          Male      28              12                30
0
##   8 no    AT          Female    31              11                60
3
##   9 no    AT          Female    16              10                10
3
## 10 no    AT          Female    57              14                30
5
## # … with 431 more rows, and 13 more variables: trust_politicians <dbl>,
## #   past_vote <chr>, immig_econ <dbl>, immig_culture <dbl>,
## #   country_attach <dbl>, religion <chr>, climate_change <dbl>,
## #   imp_tradition <dbl>, imp_equality <dbl>, income <dbl>,
## #   eu_integration <dbl>, trade_union <fct>, unemployed <fct>
```

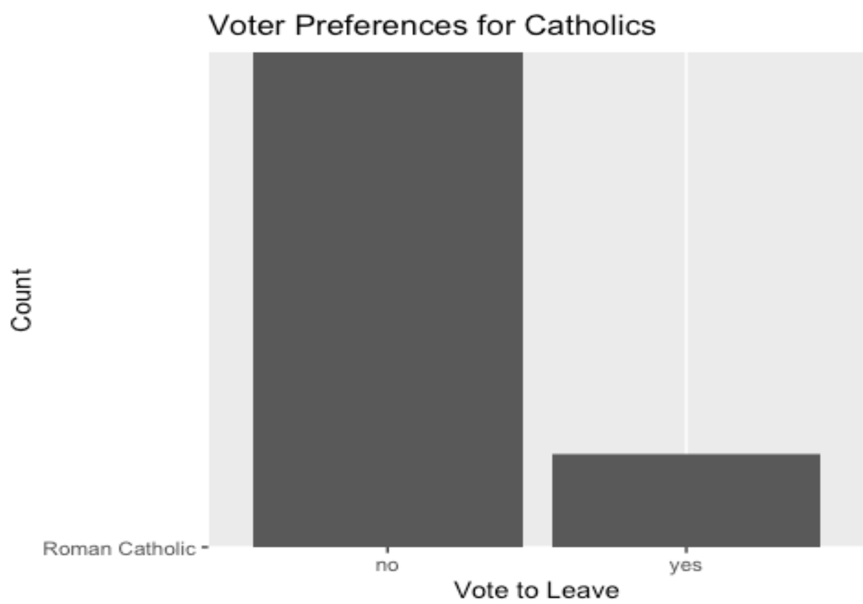
Now lets visualise the split in voter preferences for Muslims.

```
library(ggplot2)
ggplot(data = Islamic_view, aes(x = leave, y = "Islamic")) +
    geom_col() + ggtitle("Voter Preferences for Muslims") + xlab("Vote to
Leave") + ylab("Count")
```




We are going to do exactly the same process as did above for the Catholic segementation:

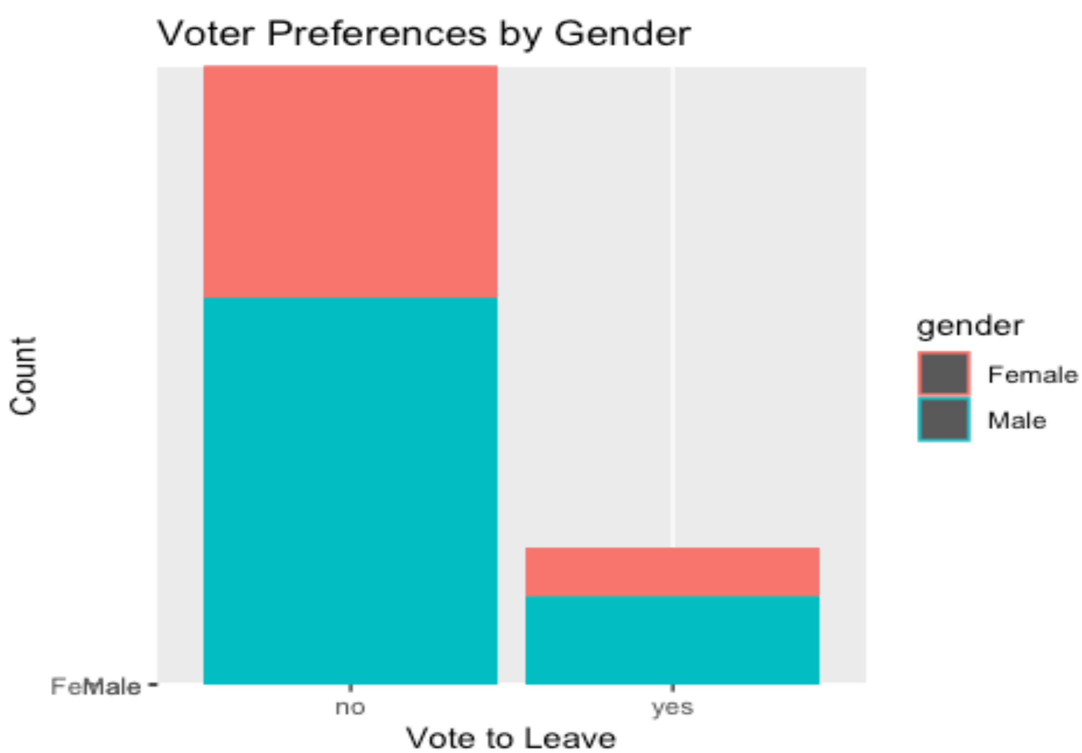
```
Catholic_view <- ess %>%
  filter(religion == "Roman Catholic")

Catholic_view

## # A tibble: 9,210 x 20
##    leave country_code gender   age years_education news_consumption
trust_people
##    <fct> <chr>        <chr>  <dbl>           <dbl>            <dbl>
<dbl>
##  1 no    AT           Female    68              13               30
5
##  2 no    AT           Female    65              13               60
3
##  3 no    AT           Female    44              17               45
7
##  4 no    AT           Female    41              16               60
5
##  5 no    AT           Female    57               9               30
2
##  6 yes   AT           Male      50              12               15
6
##  7 no    AT           Female    58               4               20
1
##  8 no    AT           Male      51              12               30
6
##  9 no    AT           Male      47              20               60
7
## 10 yes   AT           Male      45              12               35
9
## # … with 9,200 more rows, and 13 more variables: trust_politicians <dbl>,
## #   past_vote <chr>, immig_econ <dbl>, immig_culture <dbl>,
## #   country_attach <dbl>, religion <chr>, climate_change <dbl>,
## #   imp_tradition <dbl>, imp_equality <dbl>, income <dbl>,
## #   eu_integration <dbl>, trade_union <fct>, unemployed <fct>

library(ggplot2)
ggplot(data = Catholic_view, aes(x = leave, y = "Roman Catholic")) +
    geom_col() + ggtitle("Voter Preferences for Catholics") + xlab("Vote to
Leave") + ylab("Count")
```




*Gender and the European Union*

After exploring the relgious groups views on Europe, it is worth considering if there is any variation between genders.

```
ggplot(data = ess, aes(x = leave, y = gender, col = gender)) +
geom_col() + ggtitle("Voter Preferences by Gender") + xlab("Vote to Leave") +
ylab("Count")
```



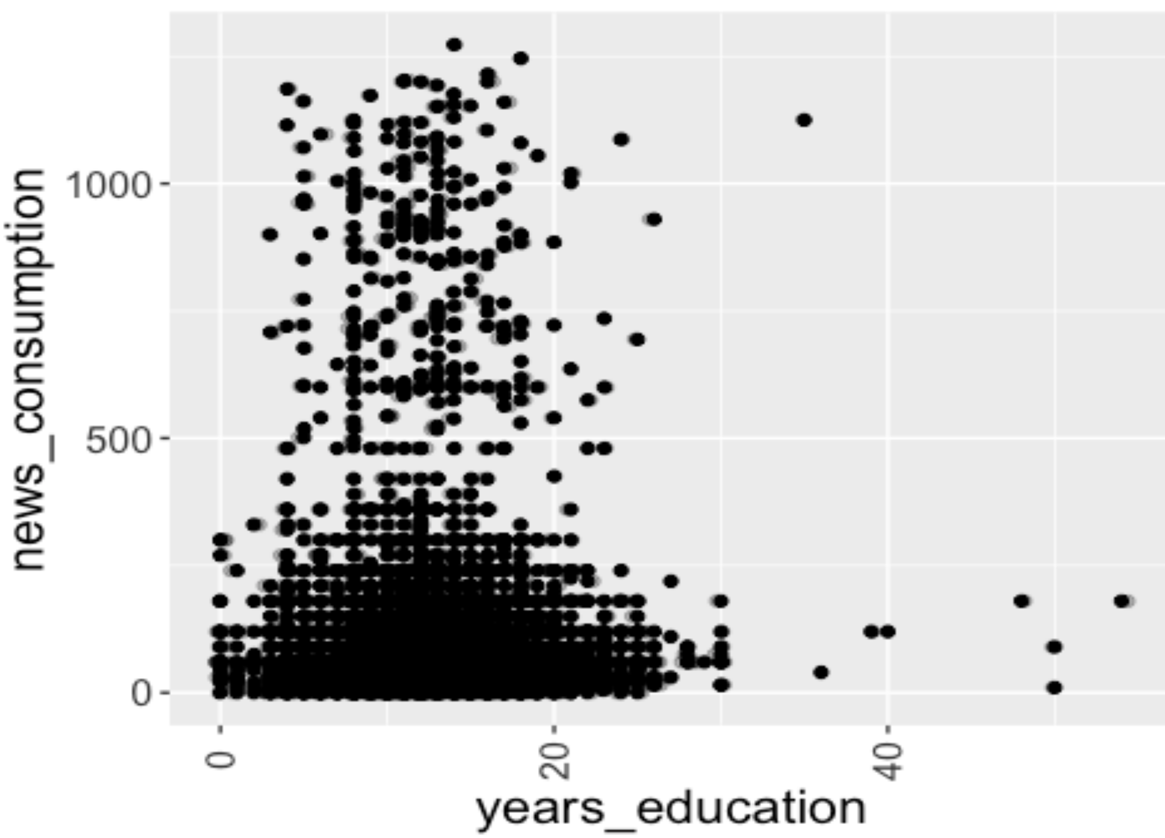
There is not much variation.

We will now analyse some of the other featurs in this dataset. Lets take a quick look at the dataset to remind ourselves of these features.

```
colnames(ess)
```

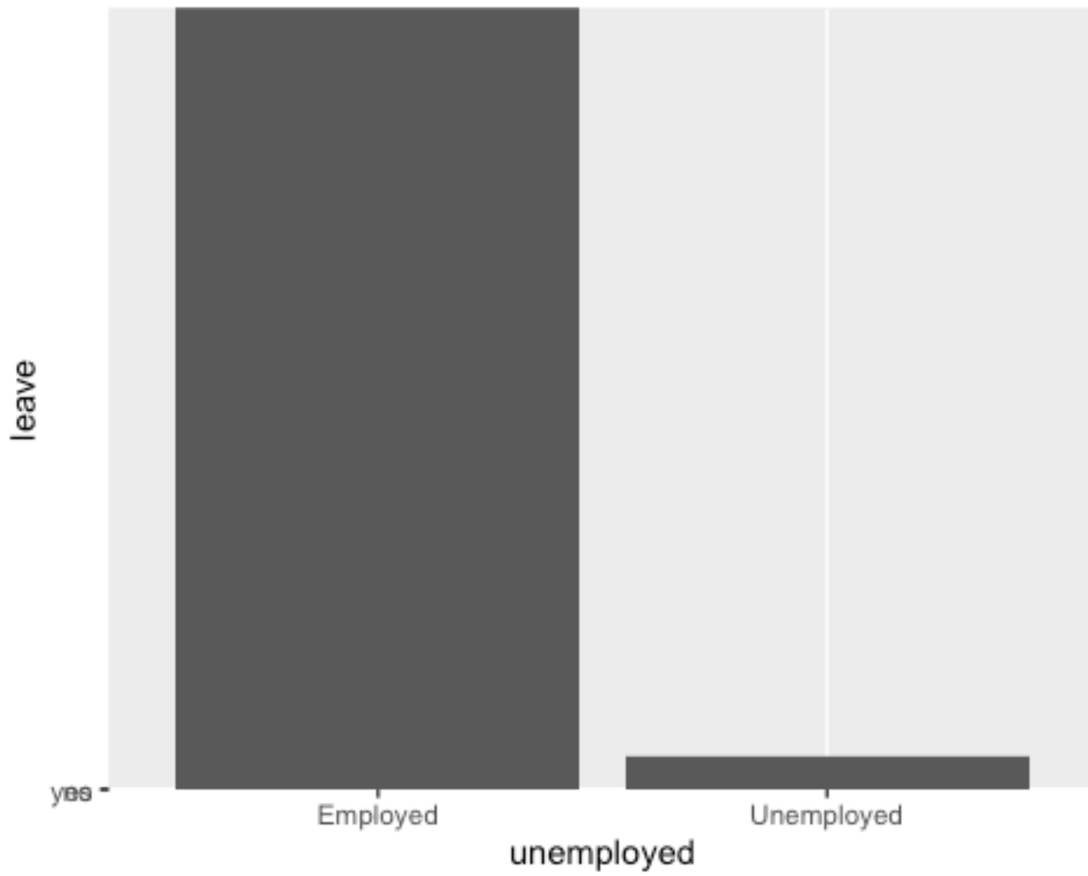
```
##  [1] "leave"             "country_code"      "gender"
##  [4] "age"               "years_education"   "news_consumption"
##  [7] "trust_people"      "trust_politicians" "past_vote"
## [10] "immig_econ"        "immig_culture"     "country_attach"
## [13] "religion"          "climate_change"    "imp_tradition"
## [16] "imp_equality"      "income"            "eu_integration"
## [19] "trade_union"       "unemployed"
```

```
grey_theme <- theme(axis.text.x = element_text(colour="grey20", size = 12,
                                                angle = 90, hjust = 0.5,
                                                vjust = 0.5),
                    axis.text.y = element_text(colour = "grey20", size = 12),
                    text=element_text(size = 16))

ggplot(ess, aes(x = years_education, y = news_consumption)) + geom_point() +
grey_theme + geom_jitter(alpha = 0.3)
```

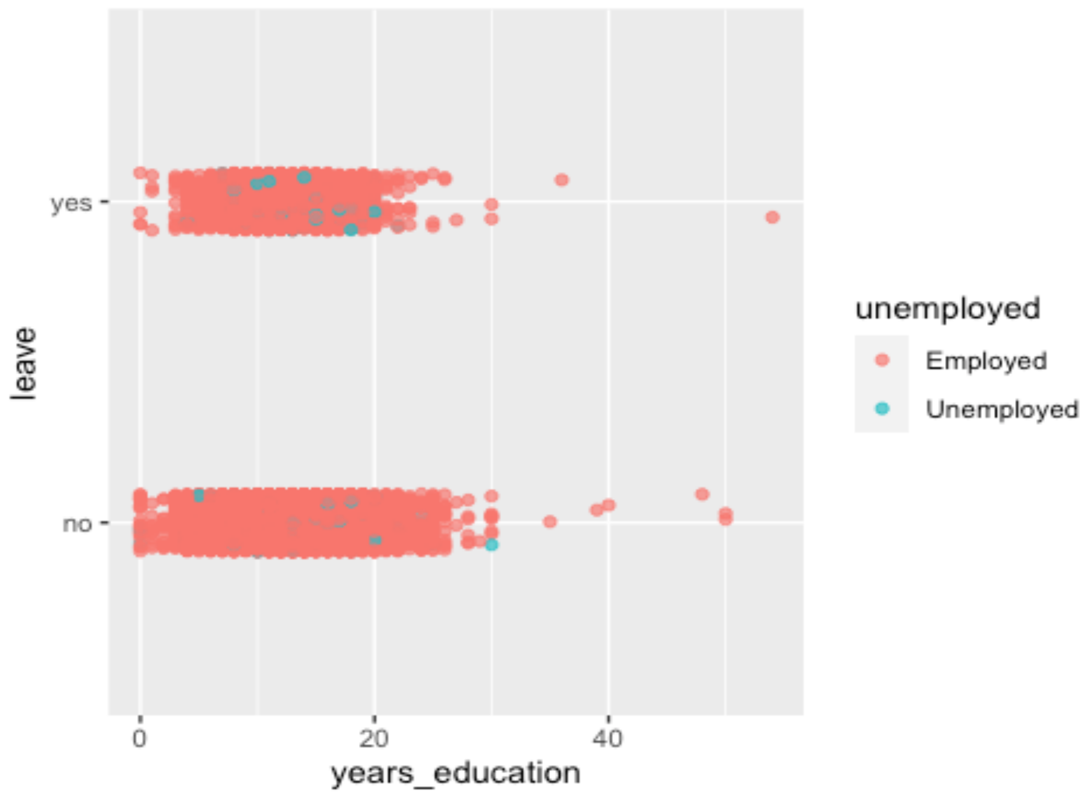


Let's do some more plotting. We should be able to compare leave votes by trade union membership and employment status.

```
#Leave by employment status
ggplot(data = ess, aes(x = unemployed, y = leave)) +
  geom_col()
```




```
#Visualization 2
library(ggplot2)
  ggplot(ess,aes(x = years_education, y = leave, color = unemployed)) +
    geom_jitter(width = 0, height = 0.09, alpha = 0.7)
```




```r
#Member of trade union?
ggplot(data = ess, aes(x = trade_union, y = leave)) +
  geom_col()
```

## Data Partition

Using the caret package, we will split the data into training and test sets. This is a critical component of machine learning approaches.

```
#load caret package
library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

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

#set the random seed
set.seed(123)
#perform train test split
training.samples <- ess$leave %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data  <- ess[training.samples, ]

## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

test.data <- ess[-training.samples, ]


head(train.data)

## # A tibble: 6 x 20
##   leave country_code gender   age years_education news_consumption
trust_people
##   <fct> <chr>        <chr>  <dbl>           <dbl>            <dbl>
<dbl>
## 1 no    AT           Female    68              13               30
5
## 2 no    AT           Female    20              13               30
5
## 3 no    AT           Female    65              13               60
3
## 4 no    AT           Female    41              16               60
5
## 5 no    AT           Female    57               9               30
2
```

```
## 6 yes   AT           Male      50              12                15
6
## # … with 13 more variables: trust_politicians <dbl>, past_vote <chr>,
## #   immig_econ <dbl>, immig_culture <dbl>, country_attach <dbl>,
## #   religion <chr>, climate_change <dbl>, imp_tradition <dbl>,
## #   imp_equality <dbl>, income <dbl>, eu_integration <dbl>, trade_union
<fct>,
## #   unemployed <fct>
```

## Modelling

Lets see what features are the best predictors of EU voter preferences. We will first test trade union membership as a predictor in voting leave in EU elections.

Throughout the modelling we will specify the family argument as binomial. This is because we are trying to predict the odds of an event taking place. Binomial logistic regression is a particular type of logistic regression in which the dependent variable y is a discrete random variable that takes on values such as 0, 1, 5, 67 etc. Each value represents the number of 'successes' observed in m trials. Thus y follows the binomial distribution.

### Model 1: Voter preferences by membership of trade union

```
#Logistic Regression Modelliing
library(aod)
#Model 1
logit_M1 <- glm(leave ~ trade_union + years_education + country_attach +
eu_integration, data = train.data, family = binomial(link = "logit"))
screenreg(logit_M1)

##
## ===============================
##                    Model 1
## -------------------------------
## (Intercept)            1.29 ***
##                       (0.15)
## trade_unionMember      0.14 *
##                       (0.06)
## years_education       -0.06 ***
##                       (0.01)
## country_attach        -0.07 ***
##                       (0.01)
## eu_integration        -0.37 ***
##                       (0.01)
## -------------------------------
## AIC                 8371.81
## BIC                 8408.09
## Log Likelihood     -4180.91
## Deviance            8361.81
## Num. obs.          10461
## ===============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
summary(logit_M1)

##
## Call:
## glm(formula = leave ~ trade_union + years_education + country_attach +
##     eu_integration, family = binomial(link = "logit"), data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6717  -0.6115  -0.4313  -0.2588   2.8846
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.29013    0.15291   8.437  < 2e-16 ***
## trade_unionMember  0.14037    0.05666   2.477   0.0132 *
## years_education   -0.05895    0.00731  -8.063 7.42e-16 ***
## country_attach    -0.06627    0.01330  -4.984 6.23e-07 ***
## eu_integration    -0.37110    0.01147 -32.364  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9752.5  on 10460  degrees of freedom
## Residual deviance: 8361.8  on 10456  degrees of freedom
## AIC: 8371.8
##
## Number of Fisher Scoring iterations: 5

probabilities <- logit_M1 %>% predict(test.data, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")
#predicted.classes
#mean(predicted.classes==test.data$leave)
```

### *Model 1: Results and Evaluation*

The first logistic regression model produced some interesting results. We can conclude that trade union membership is positively correlated with a decision to vote to leave in the EU survey. This would support Coulter (2016) who argues that trade unions as interest groups, particuarly in the UK have tended to be more sceptical of EU integration.

Paper: http://eprints.lse.ac.uk/68929/1/LEQSPaper121Coulter.pdf

We now run confidence interval tests on the model. Note that for logistic models, confidence intervals are based on the profiled log-likelihood function. We can also get CIs based on just the standard errors by using the default method.

```r
#Confidence intervals
confint(logit_M1)

## Waiting for profiling to be done...
```

```
##                          2.5 %      97.5 %
## (Intercept)         0.99064736  1.59016660
## trade_unionMember   0.02917045  0.25130241
## years_education    -0.07331647 -0.04465801
## country_attach     -0.09225988 -0.04012648
## eu_integration     -0.39372683 -0.34877369
```

```
#Confidence intervals with standard error
confint.default(logit_M1)
```

```
##                          2.5 %      97.5 %
## (Intercept)         0.99043372  1.58983156
## trade_unionMember   0.02932092  0.25142464
## years_education    -0.07327542 -0.04461886
## country_attach     -0.09232930 -0.04020774
## eu_integration     -0.39357608 -0.34862769
```

We can also test for an overall effect of rank using the wald.test function of the aod library. The order in which the coefficients are given in the table of coefficients is the same as the order of the terms in the model. This is important because the wald.test function refers to the coefficients by their order in the model. We use the wald.test function. b supplies the coefficients, while Sigma supplies the variance covariance matrix of the error terms.

```
#Wald test
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

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

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

Anova(logit_M1, type="II", test="Wald")

## Analysis of Deviance Table (Type II tests)
##
## Response: leave
##                 Df     Chisq Pr(>Chisq)
## trade_union      1    6.1378    0.01323 *
## years_education  1   65.0180  7.422e-16 ***
## country_attach   1   24.8391  6.232e-07 ***
## eu_integration   1 1047.4036  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results from the wald/chi-square tests would suggest that predictor feature variables are indeed significant. We can now proceed to the next stages of the modelling.

### Model 2: Voter preferences by employment status

For this model we will play particular attention to news consumption levels and how this interacts with trust for politicians and emotional attachment to a country.

```
#Model 2
logit_M2 <- glm(leave ~ unemployed + country_attach + news_consumption +
trust_politicians, data = train.data, family = binomial(link = "logit"))
summary(logit_M2)

##
## Call:
## glm(formula = leave ~ unemployed + country_attach + news_consumption +
##     trust_politicians, family = binomial(link = "logit"), data =
train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9669  -0.6677  -0.5570  -0.4631   2.3646
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.7112391  0.1081145  -6.579 4.75e-11 ***
## unemployedUnemployed  0.1018323  0.1264659   0.805  0.42070
## country_attach      -0.0347339  0.0125079  -2.777  0.00549 **
## news_consumption     0.0002183  0.0001921   1.136  0.25586
## trust_politicians   -0.1687207  0.0113912 -14.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9752.5  on 10460  degrees of freedom
## Residual deviance: 9502.6  on 10456  degrees of freedom
## AIC: 9512.6
##
## Number of Fisher Scoring iterations: 4

screenreg(logit_M2)

##
## ==================================
##                       Model 1
## ----------------------------------
## (Intercept)              -0.71 ***
##                          (0.11)
## unemployedUnemployed      0.10
##                          (0.13)
```

```
## country_attach           -0.03 **
##                          (0.01)
## news_consumption          0.00
##                          (0.00)
## trust_politicians        -0.17 ***
##                          (0.01)
## ---------------------------------
## AIC                    9512.64
## BIC                    9548.92
## Log Likelihood        -4751.32
## Deviance               9502.64
## Num. obs.             10461
## =================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

The feature variable unemployed is associated with a 0.10% increase in the likelihood of voting leave. However, the variable is not significant. Lets take a look at the confidence intervals for the model.

*Model 2: Results and Evaluation*

```
#Confidence intervals
confint(logit_M2)

## Waiting for profiling to be done...

##                              2.5 %        97.5 %
## (Intercept)           -0.9246509169 -0.5007279281
## unemployedUnemployed -0.1514124967  0.3449121696
## country_attach        -0.0591284230 -0.0100871450
## news_consumption      -0.0001665478  0.0005875433
## trust_politicians     -0.1911231513 -0.1464659655

#Confidence intervals with standard error
confint.default(logit_M2)

##                              2.5 %       97.5 %
## (Intercept)           -0.9231397012 -0.499338520
## unemployedUnemployed -0.1460362788  0.349700832
## country_attach        -0.0592489134 -0.010218824
## news_consumption      -0.0001582306  0.000594743
## trust_politicians     -0.1910470115 -0.146394384

#Wald test
library(aod)
wald.test(b = coef(logit_M2), Sigma = vcov(logit_M1), Terms = 1:4)

## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 499.9, df = 4, P(> X2) = 0.0
```

### Full Logistic Regression Model

Let's programme a full logistic regression that tests all the predictors.

```
full.model <- glm(leave ~., data = train.data, family = binomial)
coef(full.model)
```

```
##             (Intercept)         country_codeBE         country_codeCZ
##            1.7020095879          -0.3092015262           0.1747750796
##          country_codeDE         country_codeES         country_codeFI
##           -0.3275183568          -0.6757308019           0.6827427759
##          country_codeFR         country_codeGB         country_codeHU
##            0.3552891267           1.5806233960          -0.6114905390
##          country_codeIE         country_codeIT         country_codeLT
##           -0.5417980320          -0.1396085936          -0.5612697747
##          country_codeNL         country_codePL         country_codePT
##            0.3984928615          -0.4345066607          -0.0837655628
##          country_codeSE         country_codeSI             genderMale
##            0.8733942429           0.0808243823           0.0968388020
##                     age        years_education       news_consumption
##           -0.0029673141          -0.0323192140           0.0001075511
##            trust_people      trust_politicians            past_voteYes
##           -0.0370451053          -0.1012956732           0.1447152380
##              immig_econ          immig_culture         country_attach
##           -0.0878621077          -0.0984331231          -0.0313498080
##          religionJewish          religionOther      religionProtestant
##           -0.4342936564           0.0239497336          -0.0108224642
## religionRoman Catholic         climate_change           imp_tradition
##           -0.2174412431           0.0586688917          -0.0100576523
##            imp_equality                 income         eu_integration
##            0.0993919520          -0.2315359502          -0.3072964771
##       trade_unionMember    unemployedUnemployed
##            0.0188465400           0.2067223790
```

Country code seems to be highly correlated. Lets remove it from the dataset and try again.

```
train.data1 = train.data[!grepl("^country_code",names(train.data))]
colnames(train.data1)
```

```
##  [1] "leave"             "gender"            "age"
##  [4] "years_education"   "news_consumption"  "trust_people"
##  [7] "trust_politicians" "past_vote"         "immig_econ"
## [10] "immig_culture"     "country_attach"    "religion"
## [13] "climate_change"    "imp_tradition"     "imp_equality"
## [16] "income"            "eu_integration"    "trade_union"
## [19] "unemployed"
```

We run the full model again.

```
full.model <- glm(leave ~., data = train.data1, family = binomial)
coef(full.model)
```

```
##           (Intercept)            genderMale                   age
##          1.872281e+00          4.501286e-02         -1.455255e-03
##       years_education      news_consumption          trust_people
##         -3.351942e-02          7.959344e-05         -2.923943e-02
##     trust_politicians           past_voteYes            immig_econ
##         -8.999230e-02          1.302250e-01         -8.107462e-02
##         immig_culture        country_attach         religionJewish
##         -9.604651e-02         -4.096155e-02         -3.555712e-01
##         religionOther     religionProtestant religionRoman Catholic
##          4.681367e-02          3.740158e-01         -5.446192e-01
##        climate_change         imp_tradition          imp_equality
##          3.008668e-02          8.652864e-03          9.485439e-02
##                income        eu_integration      trade_unionMember
##         -2.334665e-01         -3.135363e-01          7.112907e-02
##   unemployedUnemployed
##          1.608285e-01
```

*Perform stepwise variable selection*

Select the most contributive variables:

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:patchwork':
##
##     area
```

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

```
step.model <- full.model %>% stepAIC(trace = FALSE)
coef(step.model)
```

```
##           (Intercept)        years_education          trust_people
##            1.97578734            -0.03055570           -0.02844275
##     trust_politicians           past_voteYes            immig_econ
##           -0.09134125             0.12794026           -0.08248544
##         immig_culture         country_attach         religionJewish
##           -0.09515157            -0.04250929           -0.38186323
##         religionOther     religionProtestant religionRoman Catholic
##            0.03501549             0.34889222           -0.57715791
##          imp_equality                 income        eu_integration
##            0.09417150            -0.22796707           -0.31293186
```

There are a number of variables that seem to be highly correlated with the voting preferences. In particular, years education, concerns about the economic impact of immigration and concerns about the cultural impact of immigration. Indeed, an ongoing academic discussion focusses on whether cultural or economic concerns about immigration are more important as predictors of support for the European Union.

### Model 3: Likelihood to vote leave by attitudes towards immigration

```
logit_M3 <- glm(leave ~ immig_econ + immig_culture, data = train.data, family
= binomial(link = "logit"))
summary(logit_M3)
```

```
##
## Call:
## glm(formula = leave ~ immig_econ + immig_culture, family = binomial(link =
"logit"),
##     data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1446  -0.6224  -0.5053  -0.3789   2.4880
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.07777    0.05990  -1.298    0.194
## immig_econ    -0.15358    0.01407 -10.919   <2e-16 ***
## immig_culture -0.14353    0.01343 -10.689   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9752.5  on 10460  degrees of freedom
## Residual deviance: 9089.9  on 10458  degrees of freedom
## AIC: 9095.9
##
## Number of Fisher Scoring iterations: 4
```

```
screenreg(logit_M3)
```

```
##
## ============================
##                 Model 1
## ----------------------------
## (Intercept)       -0.08
##                   (0.06)
## immig_econ        -0.15 ***
##                   (0.01)
## immig_culture     -0.14 ***
##                   (0.01)
## ----------------------------
## AIC             9095.92
## BIC             9117.69
## Log Likelihood  -4544.96
```

```
## Deviance         9089.92
## Num. obs.       10461
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

*Model 3: Results and Evaluation*

```
#Confidence intervals
confint(logit_M3)

## Waiting for profiling to be done...

##                    2.5 %      97.5 %
## (Intercept)   -0.1953165  0.03952158
## immig_econ    -0.1811820 -0.12604059
## immig_culture -0.1698911 -0.11725106

#Confidence intervals with standard error
confint.default(logit_M3)

##                    2.5 %      97.5 %
## (Intercept)   -0.1951683  0.03963059
## immig_econ    -0.1811443 -0.12600923
## immig_culture -0.1698431 -0.11720912
```

### Model 4: Voter preferences by number of years of education and EU integration level

```
logit_M4 <- glm(leave ~ years_education + eu_integration, data = train.data,
family = binomial(link = "logit"))
summary(logit_M4)

##
## Call:
## glm(formula = leave ~ years_education + eu_integration, family =
binomial(link = "logit"),
##     data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5177  -0.6059  -0.4398  -0.2610   2.8320
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.771884   0.102728   7.514 5.74e-14 ***
## years_education -0.055760   0.007234  -7.708 1.28e-14 ***
## eu_integration  -0.370428   0.011394 -32.510  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9752.5  on 10460  degrees of freedom
## Residual deviance: 8390.2  on 10458  degrees of freedom
```

```
## AIC: 8396.2
##
## Number of Fisher Scoring iterations: 5
```

```r
screenreg(logit_M4)
```

```
##
## ============================
##                  Model 1
## ----------------------------
## (Intercept)          0.77 ***
##                     (0.10)
## years_education     -0.06 ***
##                     (0.01)
## eu_integration      -0.37 ***
##                     (0.01)
## ----------------------------
## AIC               8396.17
## BIC               8417.94
## Log Likelihood   -4195.09
## Deviance          8390.17
## Num. obs.        10461
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

*Model 4: Results and Evaluation*

```r
#Confidence intervals
confint(logit_M4)
```

```
## Waiting for profiling to be done...
```

```
##                      2.5 %     97.5 %
## (Intercept)      0.57097347  0.9737030
## years_education -0.06997675 -0.0416181
## eu_integration  -0.39290916 -0.3482398
```

```r
#Confidence intervals with standard error
confint.default(logit_M4)
```

```
##                      2.5 %      97.5 %
## (Intercept)      0.57054046  0.97322727
## years_education -0.06993886 -0.04158204
## eu_integration  -0.39276022 -0.34809553
```

### Model 5: Voter preferences by number of years of education and attachment to the country

```r
logit_M5 <- glm(leave ~ years_education + country_attach, data = train.data,
family = binomial(link = "logit"))
summary(logit_M5)
```

```
##
## Call:
## glm(formula = leave ~ years_education + country_attach, family =
```

```
binomial(link = "logit"),
##     data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0098  -0.6532  -0.5982  -0.5156   2.4387
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.20642    0.13661  -1.511    0.131
## years_education -0.06714    0.00662 -10.142  < 2e-16 ***
## country_attach  -0.05951    0.01251  -4.756 1.97e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9752.5  on 10460  degrees of freedom
## Residual deviance: 9631.0  on 10458  degrees of freedom
## AIC: 9637
##
## Number of Fisher Scoring iterations: 4
```

```
screenreg(logit_M5)
```

```
##
## =============================
##                  Model 1
## -----------------------------
## (Intercept)         -0.21
##                     (0.14)
## years_education     -0.07 ***
##                     (0.01)
## country_attach      -0.06 ***
##                     (0.01)
## -----------------------------
## AIC                9636.99
## BIC                9658.76
## Log Likelihood    -4815.49
## Deviance           9630.99
## Num. obs.         10461
## =============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

## Predicted Probabilities

Predicted probabilities are fairly straightforward. They are probabilities that are calculated from existing probabilities, though the method does depend on the nature of the probabilities involved. For example, mutually exclusive and complementary events predict probability as the product of event probabilities, the probability of dependent and complementary events has to be calculated as a sequence. Furthermore, logistic regression is a method of predicting probabilities based on more complex variable interaction, although the regression equation itself represents odds instead of traditional slope relationships.

```
#FitStatistics
mean(train.data$leave)

## Warning in mean.default(train.data$leave): argument is not numeric or
logical:
## returning NA

## [1] NA

summary(train.data$leave)

##   no  yes
## 8614 1847

#Fitted Values and Predicted Probabilities
train.data$pps1 <- predict(logit_M1, newdata = train.data, type = "response")
train.data$evs1 <- ifelse(train.data$pps1 > 0.5, yes = 1, no = 0)
```

### Confusion Matrix: Model Accuracy

```
#Confusion matrix to find model fit - actual outcomes
confusion <- table(actual = train.data$leave, expected.value =
train.data$evs1)
confusion #Expected values for leave and remain

##       expected.value
## actual    0    1
##    no  8359  255
##    yes 1590  257

sum(diag(confusion)) / sum(confusion)

## [1] 0.8236306
```

Our model succesfully predicted with 82% accuracy.

```
#Likelihood to vote 'leave'; EU integration and 13 years of education
eu_integration_0<- predict(
  logit_M4,
  newdata = data.frame(years_education = 10, eu_integration = 5),
  type = "response"
```

```
)
eu_integration_0

##         1
## 0.1627565

#ikelihood to vote 'leave'; EU integration and 20 years of education
eu_integration_10<- predict(
  logit_M4,
  newdata = data.frame(years_education = 20, eu_integration = 5),
  type = "response"
)
eu_integration_10

##         1
## 0.1001585
```

Liklihood to vote leave is 10% given education at university level compared to around 17% with 10 years of education. However, it is clear the feature, eu_integration is far a more signficant predictor in to vote leave.

**Attachment to country by years of education**

```
#Predicted probabilities of voting leave for those who a strongly attatched
to their country
country_attatchment1 <- predict(logit_M5,
  newdata = data.frame(country_attach = 10, years_education = 20),
  type = "response"
)
country_attatchment1

##         1
## 0.1048523
```

There is a 10% chance of voting to leave with 20 years of education and high attachment to the country.

```
country_attatchment2 <- predict(logit_M5,
  newdata = data.frame(country_attach = 10, years_education = 13),
  type = "response"
)
country_attatchment2#Those less emotionally attatched more likley to vote to
remain

##         1
## 0.1578349

#Difference between the predicted probabilities
country_attatchment1 - country_attatchment2

##           1
## -0.05298259
```

On the other hand for those who have had less education but a strongly attached the country there is a 15% chance of voting to leave.

### Plots

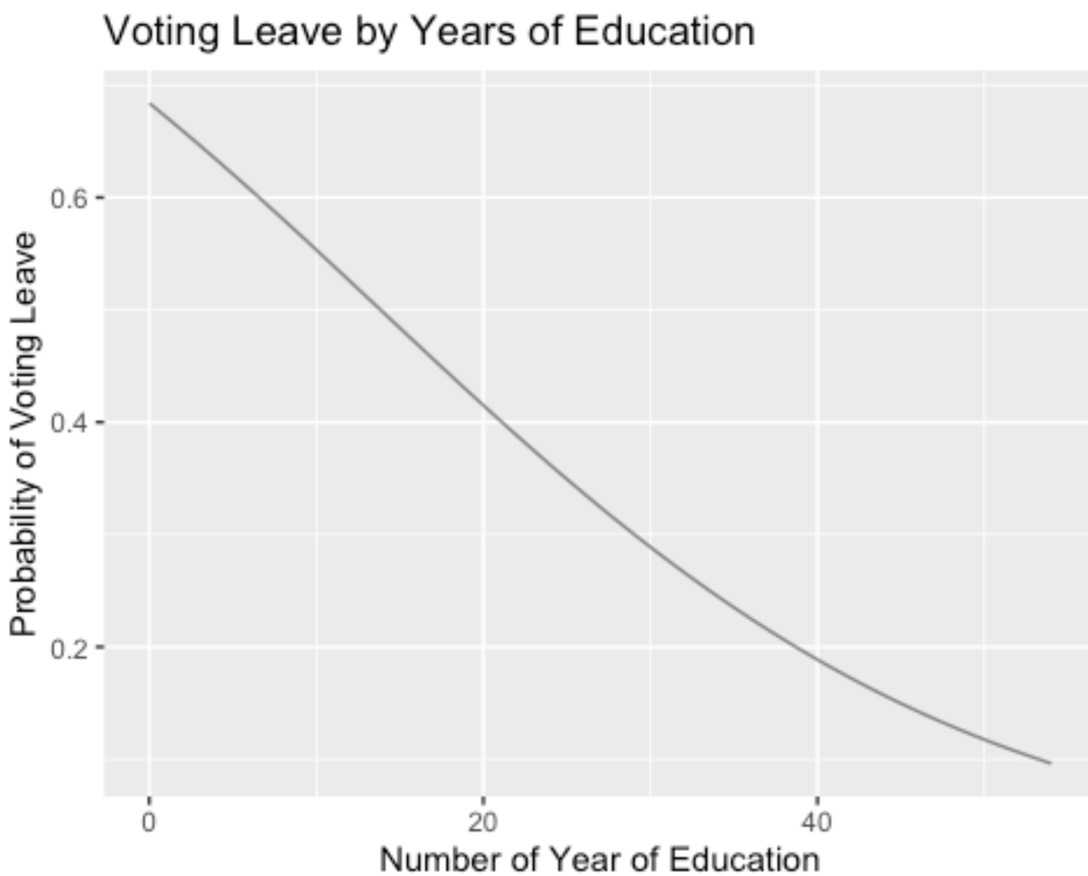
```
#Sequence years education
years_education_profiles <- data.frame(years_education = seq(from = 0, to =
54, by = .5),eu_integration = 0)
head(years_education_profiles)
```

```
##   years_education eu_integration
## 1             0.0              0
## 2             0.5              0
## 3             1.0              0
## 4             1.5              0
## 5             2.0              0
## 6             2.5              0
```

```
#create a new dataframe for years education profiles
years_education_profiles$predicted_probs <- predict(logit_M4, newdata =
years_education_profiles, type = "response")
```

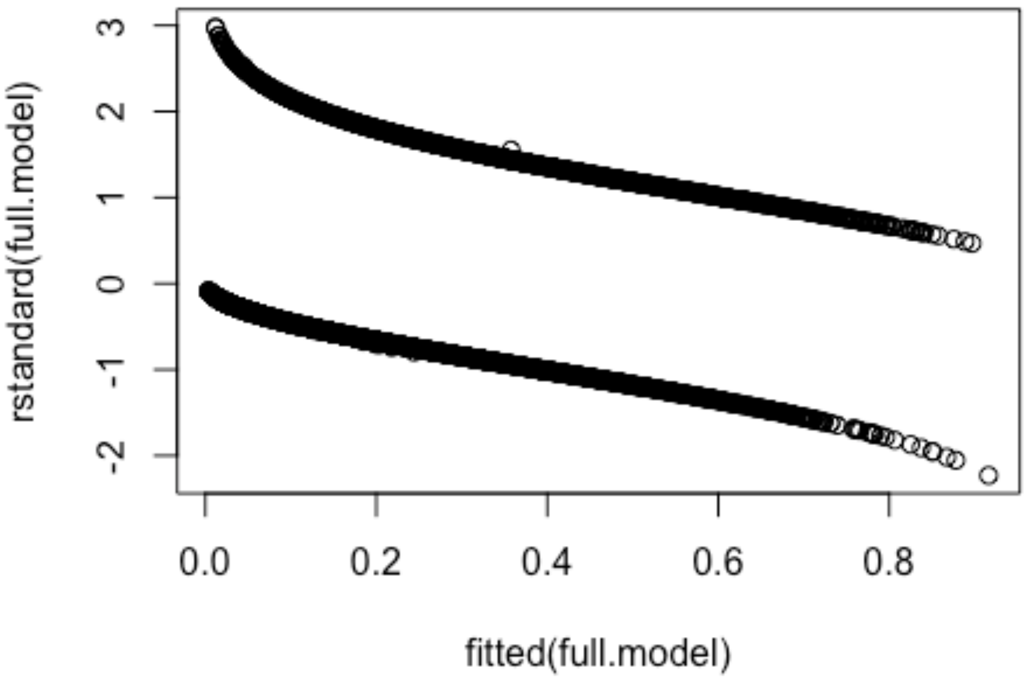
Lets now plot the relationship between years education and voter preferences.

```
#Plot 1: Voting Leave by Years of Education:
ggplot(years_education_profiles, aes(x = years_education, y =
predicted_probs)) +
  geom_line(alpha = 0.5) + ylab("Probability of Voting Leave") + xlab("Number
of Year of Education") + ggtitle("Voting Leave by Years of Education")
```




Plot the standardized residuals for the full model.

```
plot(fitted(full.model),
    rstandard(full.model))
```




## Conclusion: Findings and Future Work

The analysis above shows the complexity in predicting attitudes towards politics in general. With that said we can make some kind of conclusion that years of education and attitudes toward immigration are strong predictors of attitudes towards the EU. Going forward it would be interesting to see if we could predict which way a respondent would vote based on one or two features. We could use classification algorithms such as MNB or SVM.