Binary Models

Contents

1 Introduction

Many questions we seek to answer in social science research are binary: Will the Tories win the next election? Will Brexit help or hurt the economy? Will a country go to war? To shed light on these kinds of questions, we employ logistic regression. In this workshop we cover the theory of regression with a binary dependent variable and then turn to the practical application in R. We will estimate our own logistic regression, show by how much our model improves upon prior knowledge and illustrate our results such that an audience without statistical knowledge can interpret the results. Students should be familiar with R as well as basic concepts of statistics such as the level of measurement of a variable, the mean and linear regression.

We cover the following:

- Theory of regression with a binary dependent variable
- Estimating a logistic regression in R
- Assessing the quality of our model
- Making predictions based on our model

Last Updated: May 24, 2017 5:53 PM

2 Binary Models – Logit and Probit

Binary dependent variables are frequent in social science research...

- ... why does somebody vote or not?
- ... why does a country go to war or not?
- ... why does a legislator vote *yes* or *no*?
- ... why do some countries have the death penalty and other not?

2.1 The Linear Probability Model

The linear probability model relies on linear regression to analyze binary variables.

$$y_i = \beta_0 + \beta_1 \cdot x_{1i} + \beta_2 \cdot x_{2i} + ... + \beta_k \cdot x_{ki} + \varepsilon_i \quad (1)$$

$$Pr(y_i = 1|x_1, x_2, ...) = \beta_0 + \beta_1 \cdot x_{1i} + \beta_2 \cdot x_{2i} + ... + \beta_k \cdot x_{ki} \quad (2)$$

$$(3)$$

2.1.1 Advantages

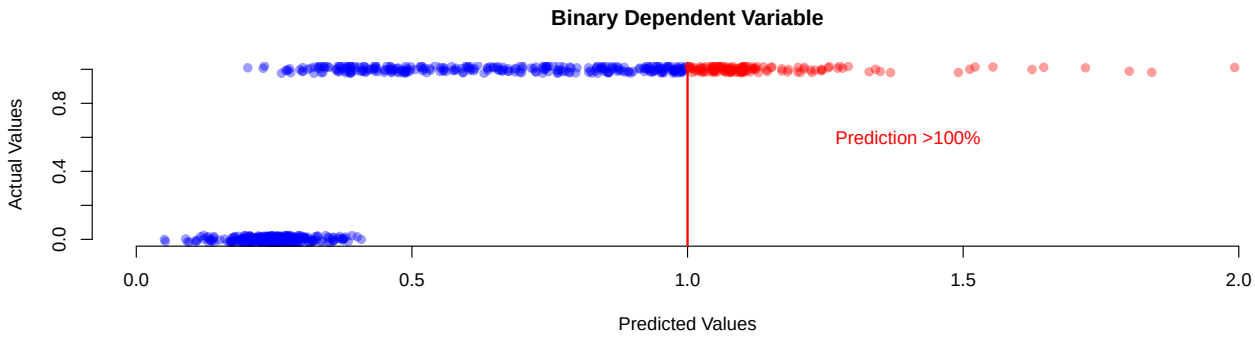
- We can use a well-known model for a new class of phenomena
- Easy to interpret the marginal effects of x variables

2.1.2 Disadvantages

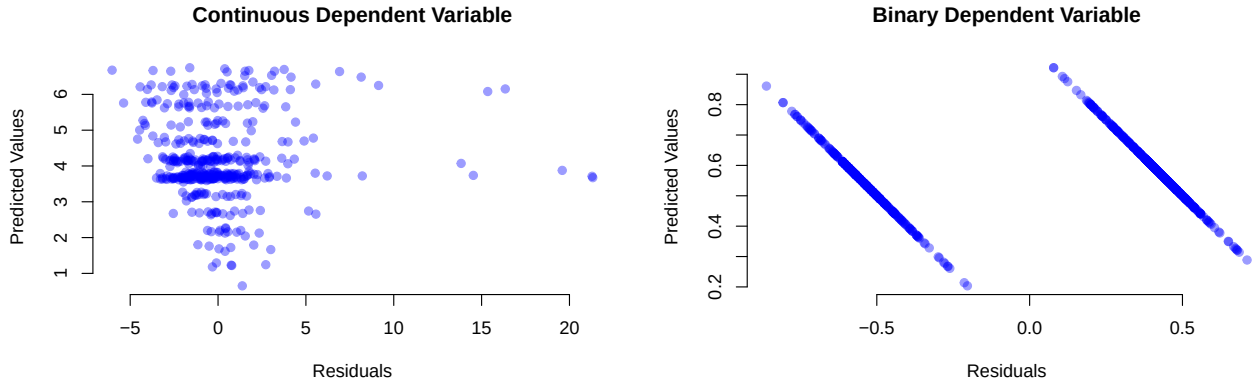
The linear model needs a continuous dependent variable, if the dependent variable is binary we run into problems:

- Predictions, $\hat{y}$, are interpreted as probability for $y = 1$
  [$\rightarrow P(y = 1) = \hat{y} = \beta_0 + \beta_1 x$, can be above 1 if x is large enough]{}
  [$\rightarrow P(y = 0) = 1 - \hat{y} = 1 - \beta_0 + \beta_1 x$, can be below 0 if x is small enough]{}
- The errors will not have a constant variance.
  [$\rightarrow$ For a given x the residual can be either (1-β_0-$\beta_1 x$) or (β_0+$\beta_1 x$)]{}
- The linear function might be wrong
  [$\rightarrow$ Imagine you buy a car. Having an additional £1000 has a very different effect if you are broke or if you already have another £12,000 for a car.]{}

Predictions can lay outside $I = [0, 1]$

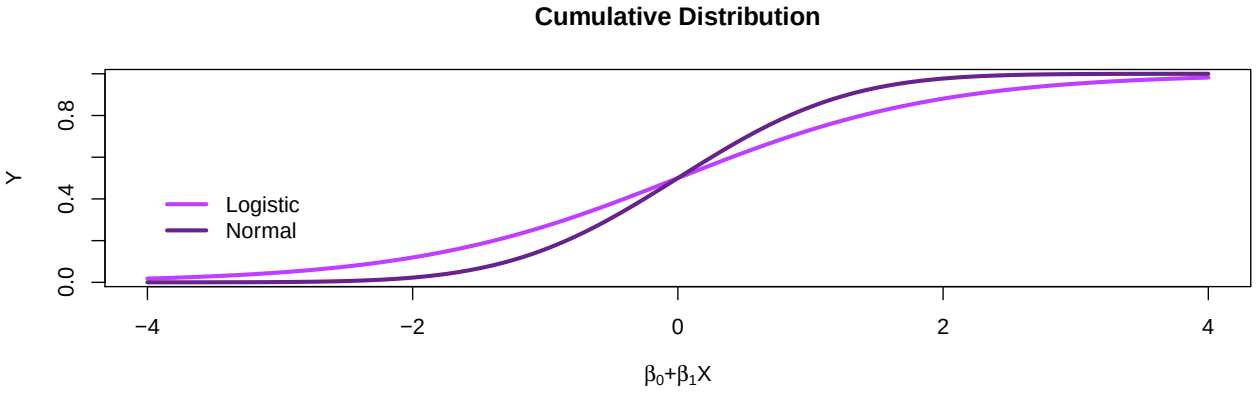


Residuals if the dependent variable is binary:



2.2 Building a Model from Probability Theory

- We want to make predictions in terms of probability
- We can have a model like this: $P(y_i = 1) = F(\beta_0 + \beta_1 x_i)$ where $F(\cdot)$ should be a function which never returns values below 0 or above 1
- There are two possibilities for $F(\cdot)$: cumulative normal (Φ) or logistic (Δ) distribution



2.3 Logit and Probit

- We now have a model where $\hat{y} \in [0, 1]$
  $\rightarrow$ All predictions are probabilities

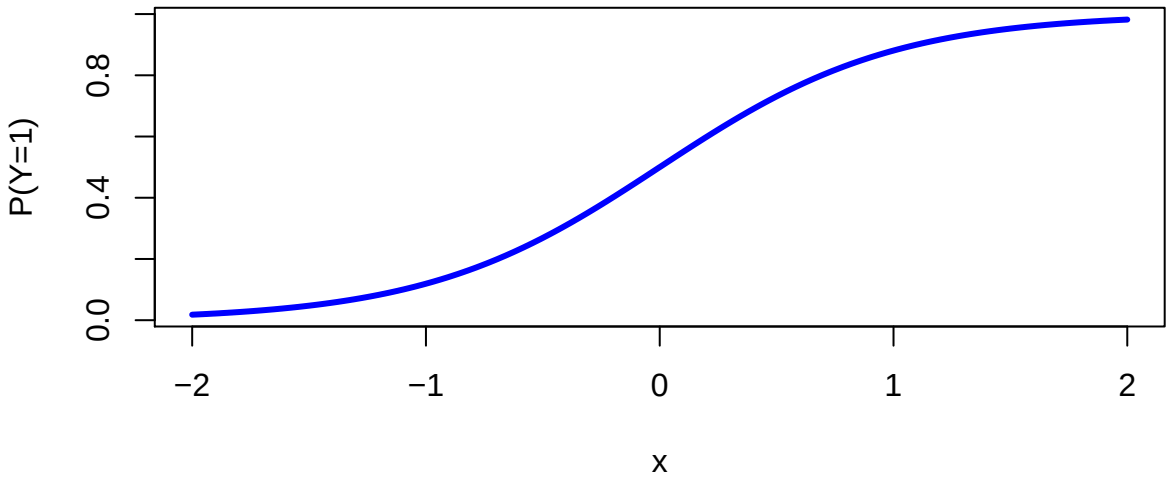
- We have two possible models to use
  [$\rightarrow$ The logit model is based on the cumulative logistic distribution (Δ)]{}
  [$\rightarrow$ The probit model is based on the cumulative normal distribution (Φ)]{}

We will use logit more often because we can write $\Delta(x) = \frac{1}{1+\exp(-x)}$,
while probit models are tricky: $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(\frac{-(x)^2}{2}) dx$

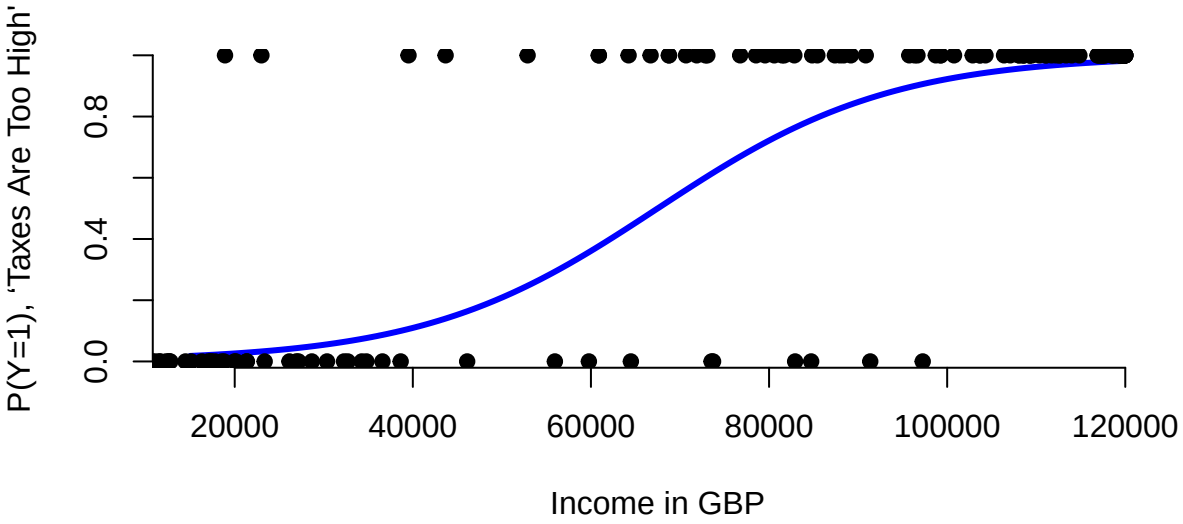
2.4 Logit Model

The logit model is then: $P(y_i = 1) = \frac{1}{1+\exp(-\beta_0 - \beta_1 x_i)}$

For $\beta_0 = 0$ and $\beta_1 = 2$ we get:

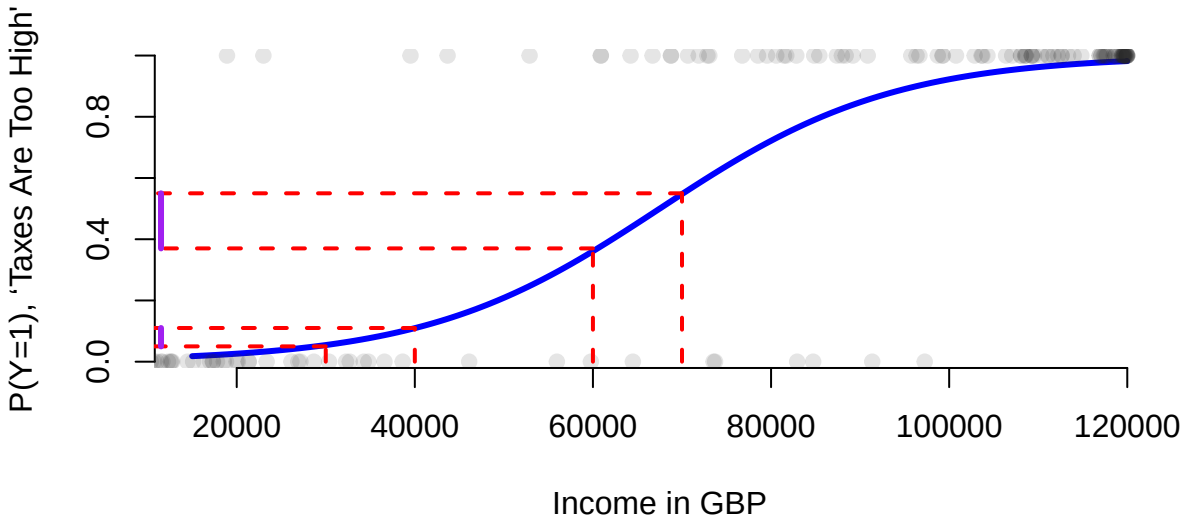


2.4.1 Logit Model: Example 1



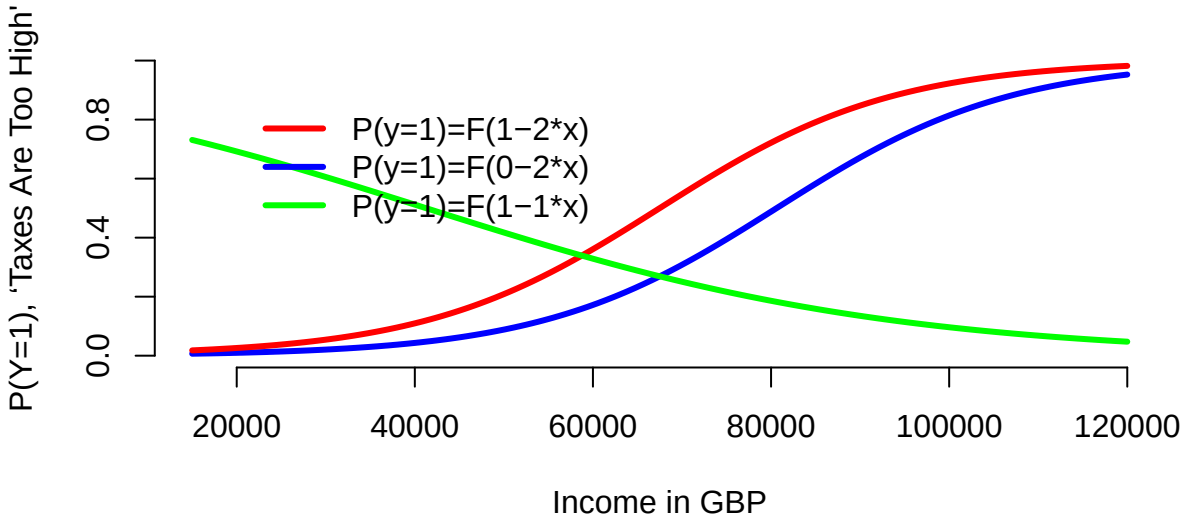
- We can make a prediction by calculating: $P(y = 1) = \frac{1}{1+\exp(-\beta_0 - \beta_1 \cdot x)}$

2.4.2 Logit Model: Example 2



- Depending on where we add £1,000 we get a different marginal effect
  [→ because of our different functional form (s-shaped)]{}

2.4.3 Logit Model: Example 3



- A positive β_1 makes the s-curve increase.
- A smaller β_0 shifts the s-curve to the right.
- A negative β_1 makes the s-curve decrease.

3 Hands-on Tutorial

Let's start by loading the required packages:

```
library(foreign)
library(texreg)
library(lmtest)
library(dplyr)
```

Clear the environment

```
rm(list = ls())
```

3.1 Application to British Election Study Dataset

We use a subset of the 2005 face-to-face British post election study to explain turnout.

Disclaimer: We use a slightly modified version of the data set. For your own research please visit the British Election Study website and download the original dataset.

You can download the dataset and codebook from the links below:

- **Dataset**
- **Codebook**

Most of the variable names in the dataset are self explanatory, but ones that need some clarification are listed below:

Variable	Description	Range
Turnout	Turnout at 2005 Brit election	No (0); Yes (1)
Gender	R's gender	1 (male); 0 (female)
LeftRightSelf	R's self placement on left-right	1 (left) - 11 (right)
CivicDutyIndex	Sum of scores on CivicDuty1-7	high values mean high civic duty
polinfoindex	Correctly answered knowledge questions	0 (low) - 8 (high)
edu*	yrs of education	binary
in.school	R still in school	binary
in.uni	R attends university	binary

3.2 Loading Data

```
bes <- read.dta("bes.dta")
```

Let's convert the `Gender` variable to a factor to help us with the interpretation of the results.

```
bes$Gender <- factor(bes$Gender, levels = c(0, 1), labels = c("Female", "Male"))
```

Now take a look at the first few observations to see what the dataset looks like

```
head(bes)
```

```
##   cs_id Turnout Vote2001 Income Age Gender PartyID Influence Attention
```

```
## 1     1       0        1      4  76 Female       1         1         8
## 2     2       1        1      5  32   Male       0         3         8
## 3     3       1       NA     NA  NA   <NA>      NA        NA        NA
## 4     4       0        1      1  35 Female       0         1         1
## 5     5       1        1      7  56   Male       0         1         9
## 6     6       1        1      4  76 Female       1         4         8
##   Telephone LeftrightSelf CivicDutyIndex polinfoindex edu15 edu16 edu17
## 1         1             7             20            7     1     0     0
## 2         1             6             15            5     0     1     0
## 3        NA            NA             NA           NA    NA    NA    NA
## 4         0             5             26            1     0     1     0
## 5         1             9             16            7     0     0     1
## 6         1             8             16            4     0     1     0
##   edu18 edu19plus in_school in_uni CivicDutyScores
## 1     0         0         0      0      -0.6331136
## 2     0         0         0      0       1.4794579
## 3    NA        NA        NA     NA              NA
## 4     0         0         0      0      -2.1466281
## 5     0         0         0      0       1.0324940
## 6     0         0         0      0       0.3658024
```

We have a number of missing values. Let's remove them from the dataset but we want to make sure we only remove observations when the variables we are interested in are missing. We'll follow the same procedure we used in week 5 for removing NA's.

While this method might seem tedious, it ensures that we're not dropping observations unnecessarily.

```
bes <- filter(bes,
              !is.na(Turnout),
              !is.na(Income),
              !is.na(polinfoindex),
              !is.na(Gender),
              !is.na(edu15),
              !is.na(edu17),
              !is.na(edu18),
              !is.na(edu19plus),
              !is.na(in_school),
              !is.na(in_uni))
```

3.3 Regression with a Binary Dependent Variable

We use the generalized linear model function `glm()` to estimate a logistic regression. The syntax is very similar to other regression functions we're already familiar with, for example `lm()` and `plm()`. The `glm()` function can be used to estimate many different models. We tell `glm()` that we've binary dependent variable and we want to use the cumulative logistic link function using the `family = binomial(link = "logit")` argument:

```
model1 <- glm(Turnout ~ Income + polinfoindex + Gender +
                edu15 + edu17 + edu18 + edu19plus + in_school + in_uni,
              family = binomial(link = "logit"),
              data = bes)

screenreg(model1)
```

```
##
## ============================
##                  Model 1
## ----------------------------
## (Intercept)        -1.14 ***
##                    (0.15)
## Income              0.03
##                    (0.02)
## polinfoindex        0.38 ***
##                    (0.02)
## GenderMale         -0.35 ***
##                    (0.08)
## edu15               0.38 ***
##                    (0.10)
## edu17               0.46 **
##                    (0.15)
## edu18               0.11
##                    (0.14)
## edu19plus           0.24 *
##                    (0.12)
## in_school           0.15
##                    (0.39)
## in_uni             -0.72 **
##                    (0.25)
## ----------------------------
## AIC              4401.20
## BIC              4464.53
## Log Likelihood  -2190.60
## Deviance         4381.20
## Num. obs.        4161
## ============================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

3.4 Model Quality

To assess the predictive accuracy of our model we will check the percentage of cases that it correctly predicts. Let's look the turnout data first. According to the codebook, `0` means the person did not vote, and `1` means they voted.

```
table(bes$Turnout)
```

```
##
##    0    1
## 1079 3082
```

If we look at the mean of Turnout we will see that is `0.74`. That means `74.07%` of the respondents said that they turned out to vote. If you predict for every respondent that they voted, you will be right for `74.07%` of the people. That is the naive guess and the benchmark for our model. If we predict more than `74.07%` of cases correctly our model adds value.

Below we will estimate predicted probabilities for each observation. That is, the probability our model assigns that a respondent will turn out to vote for every person in our data. To do so we use the `predict()` function with the following the usage syntax:

```
predict(model, type = "response")
```

Type `help(predict.glm)` for information on all arguments.

```
predicted_probs <- predict(model1, type = "response")
```

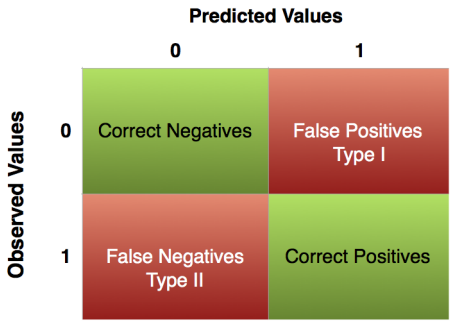
Now that we have assigned probabilities to people turning out to vote, we have to translate those into outcomes. Turnout is binary. A straightforward way would be to say: we predict that all people with a predicted probability above `50%` vote and all with predicted probabilities below or equal to `50%` abstain.

Using the `50%` threshold, we create a new variable `expected_values` that is `1` if our predicted probability is larger than `0.5` and `0` otherwise.

```
expected <- as.numeric(predicted_probs > 0.5)
```

NOTE: The condition `predicted_probs > 0.5` gives us `FASLE/TRUE`, but since our dataset uses `0/1` for the `Turnout` variable, we convert it to a numeric value using `as.numeric()` function.

All we have to do now is to compare our expected values of turnout against the actually observed values of turnout. We want our predictions to coincide with the actually observed outcomes. But we need to be aware of two types of errors we can make:



- A type I error is a false positive. It is like crying wolf when there is nothing there (we mistakenly reject a true null hypothesis). - A type II error is a false negative. It is like not noticing the wolf and finding all your sheep dead the next morning (we fail to reject a false null hypothesis).

The threshold that we set is directly related to the proportion of type I to type II errors. Increasing the threshold will reduce the number of false positives but increase the number of false negatives and vice versa. The default choice is a threshold of `0.5`. However, you may imagine situations where one type of error is more costly than the other. For example, given the cost of civil wars, a model predicting civil war may focus on reducing false negatives at the cost of a larger fraction of false positives.

We proceed by producing a table of predictions against actual outcomes. With that we will calculate the percentage of correctly predicted cases and compare that to the naive guess. We have the actually observed cases in our dependent variable (`Turnout`). The table is just a frequency table. The percentage of correctly predicted cases is simply the sum of correctly predicted cases over the number of cases.

```
observed <- bes$Turnout
```

```
outcome <- table(observed,expected)
outcome
```

```
##          expected
## observed    0    1
##        0  160  919
##        1  108 2974
```

Now let's find out how many we correctly predicted and how many we got wrong. You can manually add the numbers if you want, but it's simple enough to take the values out of the 2x2 table:

- Correct negatives are in row `1`, column `1`
- Correct positives are in row `2`, column `2`
- All others are incorrect

```
(outcome[1,1] + outcome[2,2]) / sum(outcome)
```

```
## [1] 0.7531843
```

Now let's remember what the actual turnout was:

```
table(bes$Turnout)
```

```
##
##    0    1
## 1079 3082
```

In order to calculate the naive guess, we simply divide the mode by the total number of observations

```
3082 / (1079 + 3082)
```

```
## [1] 0.7406873
```

You can see that our model outperforms the naive guess slightly. The more lopsided the distribution of your binary dependent variable, the harder it is to build a successful model.

3.4.1 Joint hypothesis testing

We will add two more explanatory variables to our model: `Influence` and `Age`. `Influence` corresponds to a theory we want to test while `Age` is a socio-economic control variable.

We want to test two theories: the rational voter model and the resource theory.

- Influence operationalises a part of the rational voter theory. In that theory citizens are more likely to turnout the more they belive that their vote matters. The variable `Influence` measures the subjective feeling of being able to influence politics.
- The variables `Income` and `polinfoindex` correspond to a second theory which states that people who have more cognitive and material resources are more likely to participate in politics.

Depending on whether the coefficients corresponding to the respective theories are significant or not, we can say something about whether these theories help to explain turnout.

The remaining variables for gender, education and the added variable `Age` are socio-economic controls. We added age because previous research has shown that political participation changes with the life cycle.

```
model2 <- glm(Turnout ~ Income + polinfoindex + Influence + Gender + Age +
                edu15 + edu17 + edu18 + edu19plus + in_school + in_uni,
              family = binomial(link = "logit"), data = bes)

screenreg(list(model1, model2))
```

```
##
## ========================================
##                  Model 1       Model 2
## ----------------------------------------
## (Intercept)        -1.14 ***     -3.90 ***
##                    (0.15)        (0.22)
## Income              0.03          0.15 ***
##                    (0.02)        (0.02)
## polinfoindex        0.38 ***      0.25 ***
##                    (0.02)        (0.02)
## GenderMale         -0.35 ***     -0.36 ***
##                    (0.08)        (0.08)
## edu15               0.38 ***     -0.34 **
##                    (0.10)        (0.11)
## edu17               0.46 **       0.36 *
##                    (0.15)        (0.16)
## edu18               0.11          0.14
##                    (0.14)        (0.15)
## edu19plus           0.24 *        0.01
##                    (0.12)        (0.13)
## in_school           0.15          1.13 **
##                    (0.39)        (0.40)
## in_uni             -0.72 **      -0.05
##                    (0.25)        (0.27)
## Influence                         0.21 ***
##                                  (0.02)
## Age                               0.05 ***
##                                  (0.00)
## ----------------------------------------
## AIC              4401.20       4003.90
## BIC              4464.53       4079.90
## Log Likelihood  -2190.60      -1989.95
## Deviance         4381.20       3979.90
## Num. obs.        4161          4161
## ========================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

The variables corresponding to the resource theory are `Income` and `polinfoindex`. In our model 2 (the one we interpret), both more income and more interest in politics are related to a higher probability to turn out to vote. That is in line with the theory. Interestingly, income was previously insignificant in model 1. It is quite likely that model 1 suffered from omitted variable bias.

As the rational voter model predicts, a higher subjective probability to cast a decisive vote, which we crudly proxy by the feeling to be able to influence politics, does correspond to a higher probability to vote as indicated by the positive significant effect of our `Influence` variable.

We will test if model 2 does better at predicting turnout than model 1. We use the likelihood ratio test. This is warranted because we added two variables and corresponds to the F-test in the OLS models estimated before. You can see values for the logged likelihood in the regression table. You cannot in general say whether a value for the likelihood is small or large but you can use it to compare models that are based on the same sample. A larger log-likelihood is better. Looking at the regression table we see that the log-likelihood is larger in model 2 than in model 1.

We use the `lrtest()` function from the `lmtest` package to test whether that difference is statistically significant. The syntax is the following:

```
lrtest(model1, model2)
```

```
## Likelihood ratio test
##
## Model 1: Turnout ~ Income + polinfoindex + Gender + edu15 + edu17 + edu18 +
##     edu19plus + in_school + in_uni
## Model 2: Turnout ~ Income + polinfoindex + Influence + Gender + Age +
##     edu15 + edu17 + edu18 + edu19plus + in_school + in_uni
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1  10 -2190.6
## 2  12 -1990.0  2 401.3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value in the likelihood ratio test is smaller than `0.05`. Therefore, the difference between the two log-likelihoods is statistically significant which means that model 2 is better at explaining turnout.

Now let's see if AIC and BIC agree.

```
AIC(model1, model2)
```

```
##        df      AIC
## model1 10 4401.200
## model2 12 4003.901
```

```
BIC(model1, model2)
```

```
##        df      BIC
## model1 10 4464.535
## model2 12 4079.903
```

According to both AIC and BIC, model 2 has a better fit.

4 Exercises

- **Dataset**
- **Codebook**

1. Select at least 4 predictors (including `age` but excluding `exper`) to explain women in the labor force and provide theoretical justification for your selection.
2. Estimate a model using the predictors you selected, present your findings and provide an assessment of the model quality.
3. Does the addition of `exper` to your model improve its predictive performance?
4. Based on your model, what's the probability of a 40 year old woman with 12 years of experience to be in the labor force? Make sure to include a confidence interval for your prediction.
5. Compare the effects of experience in the labor market on the outcome variable under two hypothetical scenarios:
    - A woman with age in the 25th percentile
    - A woman with age in the 75th percentile
6. Provide a visualization to illustrate your findings.

Reference: Mroz, T.A. (1987): "The Sensitiviy of an Empirical Model of Married Women's Hours of Work to Economic and Statistical Assumptions", Econometrica, 55, 765-799.