Tutorial 2: Logit Regression

May 13, 2024

In this problem set, we will first review the main concepts of logarithms and probability. Next, we will then use logit regression to analyze the replication data of a recently published article. The key objectives are to help you understand and carry out logit regression in `R` and interpret the results using odds ratio and predicted probability.

Before you start, please download the dataset from GitHub and import it into `RStudio`. If you would like to submit this problem set, please complete the questions at the end.

1 A Recap on Logarithm

A **logarithm** is the opposite of a **power**. In other words, if we take a logarithm of a number, we undo an exponentiation. To put it simply

- We raise **base** to a specified **power** to derive the result of an **exponentiation**.
- Logarithm means we use the result of an **exponentiation** and a given **base** to derive the corresponding **power**.

We can start with simple example. If we take the base $b = 2$ and raise it to the power of $k = 3$, we can express this procedure as 2^3. We can carry out the calculation easily in `R` as follows.

Suppose we call the result as c as follows $2^3 = c$. We can use the rules of exponentiation to calculate that the result is

$$c = 2^3 = 8.$$

```r
c <- 2^3
c
```

```
[1] 8
```

Now, say we do NOT know the value of k; instead, we know that the base is $b = 2$ and the final result of the exponentiation is $c = 8$ such that $2^k = 8$.

From the above calculation, we already know that $k = 3$. But, what if now the result of the exponentiation was $c = 4$, so we need to solve $2^k = 4$?

A logarithm is a function that does all this work for us. We define a logarithm with the base 2 to be the solution to the problems above. Log with a base of 2 is defined so that

$$\log_2 c = k,$$

where k refers to the power and c refers to the result of exponentiation when we raise 2 to the power of k.

For the purpose of logit regression, we will use a special type of logarithm with base e:

$$\log_e c = k,$$

which sometimes can be written as

$$\ln c = k.$$

To obtain c, the results of rasing e to the power of k, we will need to take the exponent of k as follows

$$e^k = c.$$

In `R`, we have functions such as `log()` and `exp()` to take the log and exponent with base e.

```r
log(4, base=2) # log of 4 with base 2
```

```
[1] 2
```

```r
log(4) # log of 4 with base e
```

```
[1] 1.386294
```

```r
?log # see what log() does
```

To derive odds ratio, we will need the *quotient* rule:

$$\log\left(\frac{a}{b}\right) = \log a - \log b$$

.

2 Variables

We will study the replication data of the following 2022 report released by the Pew Research Center: "*AI and Human Enhancement: Americans' Openness Is Tempered by a Range of Concerns.*"[1]

Based in Washington DC, the **Pew Research Center** is a leading research institute that informs the public about the issues, attitudes and trends around the world. They conduct public opinion polling, demographic research, content analysis and other data-driven social science research.

The data analyzed in this report was from the **American Trends Panel Wave 99** (November 1-7, 2021). You can download the data here: https://www.pewresearch.org/dataset/american-trends-panel-wave-99/ by creating an account for yourself. You are also encouraged to sign up for their newsletters to stay posted about the most cutting-edge survey research.

According to their "Topline" document, in total 10,260 US adults were surveyed. We will consider some of the following explanatory variables that can help explain the attitudes of Ameircans towards use of facial recognition technology by police. On GitHub you can find a separate codebook in the `.xlsx` file format.

- `FACEREC2_W99` – a categorical variable to record the responses for the following question: "*Do you think the widespread use of facial recognition technology by police would be a...?*"
- `F_EDUCCAT` – a categorical variable to record respondents' education level.
- `F_PARTY_FINAL` – a categorical variable to record respondents' party ID.
- `F_GENDER` – a binary variable for record respondents' gender ID.

Let's read in the data and check that all of our variables are as described.

```r
library(haven)
library(tidyverse)
library(stargazer)
```

We will use `read_spss` from the `haven` package this time because the original dataset `ATP W99.sav` is in the data file formate for SPSS.

```r
dta <- read_spss("ATP W99.sav")
```

[1]You can read the report here: https://www.pewresearch.org/internet/2022/03/17/ai-and-human-enhancement-americans-openness-is-tempered-by-a-range-of-concerns/.

2.1 Select Variables

Let's start by selecting the variables we might need. As you can see, I have chosen more variables here. Find what these addiional variables refer to from the codebook.

```
dta_sel <- dta %>%
  dplyr::select(QKEY,
                FACEREC2_W99,
                F_METRO, F_AGECAT, F_GENDER, F_EDUCCAT, F_HISP, F_YEARSINUS,
                F_RACECMB, F_RACETHNMOD, F_RELIG, F_PARTY_FINAL, F_INC_SDT1,
                F_REG, F_IDEO,
                WEIGHT_W99, F_CREGION, F_CDIVISION) %>%
  unique()
```

We can use `summary()` to get a snapshot of the summary statistics of each variable.

```
summary(dta_sel)
```

```
      QKEY              FACEREC2_W99        F_METRO          F_AGECAT     
 Min.   :1.003e+05   Min.   : 1.000   Min.   :1.000   Min.   : 1.000  
 1st Qu.:2.017e+11   1st Qu.: 1.000   1st Qu.:1.000   1st Qu.: 2.000  
 Median :2.018e+11   Median : 2.000   Median :1.000   Median : 3.000  
 Mean   :1.734e+11   Mean   : 2.061   Mean   :1.121   Mean   : 3.135  
 3rd Qu.:2.019e+11   3rd Qu.: 3.000   3rd Qu.:1.000   3rd Qu.: 4.000  
 Max.   :2.021e+11   Max.   :99.000   Max.   :2.000   Max.   :99.000  
                     NA's   :5107                                     
    F_GENDER        F_EDUCCAT          F_HISP        F_YEARSINUS    
 Min.   : 1.00   Min.   : 1.000   Min.   : 1.00   Min.   : 1.000  
 1st Qu.: 1.00   1st Qu.: 1.000   1st Qu.: 2.00   1st Qu.: 1.000  
 Median : 2.00   Median : 1.000   Median : 2.00   Median : 1.000  
 Mean   : 1.78   Mean   : 1.964   Mean   : 2.16   Mean   : 1.675  
 3rd Qu.: 2.00   3rd Qu.: 2.000   3rd Qu.: 2.00   3rd Qu.: 1.000  
 Max.   :99.00   Max.   :99.000   Max.   :99.00   Max.   :99.000  
                                                                  
   F_RACECMB       F_RACETHNMOD       F_RELIG        F_PARTY_FINAL   
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.000  
 Median : 1.000   Median : 1.000   Median : 2.000   Median : 2.000  
 Mean   : 3.324   Mean   : 2.974   Mean   : 4.943   Mean   : 3.093  
 3rd Qu.: 1.000   3rd Qu.: 2.000   3rd Qu.: 9.000   3rd Qu.: 3.000  
 Max.   :99.000   Max.   :99.000   Max.   :99.000   Max.   :99.000  
```

```
   F_INC_SDT1         F_REG            F_IDEO          WEIGHT_W99
 Min.    : 1.000   Min.    : 1.000   Min.    : 1.000   Min.   :0.004226
 1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.:0.377472
 Median : 6.000   Median : 1.000   Median : 3.000   Median :0.598619
 Mean    : 9.593   Mean    : 1.394   Mean    : 4.417   Mean   :1.000000
 3rd Qu.: 9.000   3rd Qu.: 1.000   3rd Qu.: 4.000   3rd Qu.:1.022797
 Max.    :99.000   Max.    :99.000   Max.    :99.000   Max.   :8.524234
                   NA's    :396
   F_CREGION       F_CDIVISION
 Min.   :1.000   Min.   :1.000
 1st Qu.:2.000   1st Qu.:3.000
 Median :3.000   Median :5.000
 Mean   :2.687   Mean   :5.205
 3rd Qu.:3.000   3rd Qu.:7.000
 Max.   :4.000   Max.   :9.000
```

2.2 Read Variables

When you read `.sav` into `R`, a lot of information is in fact hidden in the background. Let's retrieve the details of a variable carefully.

```
attr(dta_sel$FACEREC2_W99, "label")
```

```
[1] "FACEREC2_W99. Do you think the widespread use of facial recognition technology by pol
```

```
attr(dta_sel$FACEREC2_W99, "labels")
```

```
Good idea for society  Bad idea for society              Not sure
                    1                     2                     3
              Refused
                   99
```

The function `attr()` is quite useful, as it allows us to check the label of individual variables (i.e., what this variable means) and the labels used to code the responses (i.e., what values this variable includes and what these values refer to respectively).

Once again, it is also a good idea to use `table()` to check the distribution of responses.

```
table(dta_sel$FACEREC2_W99)
```

```
   1    2    3   99 
2404 1373 1362   14 
```

2.3 Recode Variables

As you can see, the responses to the question about the use of facial recognition are quite interesting (and puzzling) for a number of issues. First, a fair number of citizens said they were "not sure". When you take a look at the data more carefully, moreover, this variable contains tons of `NA`s.

We will discuss how to deal with different kinds of "no responses" more carefully in class, but for the time being let's assume those who did not say "Good idea for society" as if they were saying "Bad idea for society."

Note we will have to treat those who did not have a response as if they were saying AI is bad. We will deconstruct the code below in class.

```
dta_sel$yes_fr <- ifelse(dta_sel$FACEREC2_W99 == 1, 1, 0)
dta_sel$yes_fr[is.na(dta_sel$yes_fr) == T] = 0
summary(dta_sel$yes_fr)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0000  0.2343  0.0000  1.0000 
```

Similarly, let's recreate a binary variable `highedu` for those having a college degree or above.

```
dta_sel$highedu <- ifelse(dta_sel$F_EDUCCAT == 1, 1, 0)
summary(dta_sel$highedu)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  1.0000  0.5091  1.0000  1.0000 
```

And we can use a variable `female` to identify those who indicate female as their primary gender identity.

```r
dta_sel$female <- ifelse(dta_sel$F_GENDER == 2, 1, 0)
summary(dta_sel$female)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  1.0000  0.5485  1.0000  1.0000 
```

Given that we are studying the United States, policing has become quite politicized in recent years, especially after the BLM movement. Let's create a variable to identify the Democrats.

```r
dta_sel$democrat <- ifelse(dta_sel$F_PARTY_FINAL == 2, 1, 0)
summary(dta_sel$democrat)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0000  0.3247  1.0000  1.0000 
```

3 Logit Regression

Our primary task now is to use function `glm()` and estimate the correlates of *people's support for the use of facial recognition.*

- First, we will fit the logit model *with no predictor* and use the estimated intercept to uncover the probability of respondents supporting the use of facial recognition.
- Next, we will fit the logit model with one predictor and interpret the estimated slope using odds-ratio and (if there is time) the predicted probability using `predict()`.

3.1 Intercept Only

We will start with a simple model with no predictor. This exercise will help you become familiar with the differences between **probability**, **odds** and **log odds**.

Quickly recall the logit link function transforms a **probability** into **log odds** as follows:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right),$$

where p is the **probability** of respondents supporting the use of facial recognition by police.

If we only include the intercept in the model (i.e., no predictor), then our model is as follows:

$$\log\left(\frac{p}{1-p}\right) = \alpha + \epsilon.$$

The estimated intercept will be the estimated *log odds* of people supporting facial recognition.

Let's go ahead and fit the model. Use the `summary()` function to see the output.

```r
mod_intercept <- glm(yes_fr ~ 1,
                     data = dta_sel,
                     family = binomial)
summary(mod_intercept)
```

```

Call:
glm(formula = yes_fr ~ 1, family = binomial, data = dta_sel)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) -1.18414    0.02331   -50.8   <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: 11172  on 10259  degrees of freedom
Residual deviance: 11172  on 10259  degrees of freedom
AIC: 11174

Number of Fisher Scoring iterations: 4
```

Looking at the output, we know $\hat{\alpha} = -1.18414$. Let's use `coef()` to extract the coefficient.

```
coefs <- coef(mod_intercept) # extracting log odds
coefs
```

```
(Intercept)
  -1.184144
```

Next, we can take the exponent of the log odds to get the *odds* of people against abortion (i.e., use `exp()`). **Rule of thumb: use `exp()` to turn log odds into odds.**

```
exp(coefs) # calculating odds
```

```
(Intercept)
  0.3060081
```

Third, we can compute the *probability* of people against abortion. From the previous step, since the odds is

$$\frac{p}{1-p} = 0.3060081,$$

we know

$$p = \frac{0.3060081}{1 + 0.3060081} = 0.234308.$$

```
exp(coefs)/(1+exp(coefs)) # calculating p
```

```
(Intercept)
   0.234308
```

We can show the **confidence interval** of the estimated log odds, using the `confint()` function. These are the upper and lower bounds of the 95% confidence intervals of the log odds, and you can use them to derive the CIs of the estimated probability using the same logic above.

```r
coefs_ci <- confint(mod_intercept)
coefs_ci
```

```
    2.5 %    97.5 %
-1.230014 -1.138644
```

We can also use the upper and lower bounds of log odds to calculate the upper and lower bounds of estimated odds and probabilities.*

```r
coefs_ci <- confint(mod_intercept)
coefs_ci # CIs of log odds
```

```
    2.5 %    97.5 %
-1.230014 -1.138644
```

```r
exp(coefs_ci) # CIs of odds
```

```
    2.5 %    97.5 %
0.2922886 0.3202531
```

```r
exp(coefs_ci)/(1+exp(coefs_ci)) # CIs of probability
```

```
    2.5 %    97.5 %
0.2261790 0.2425695
```

We can verify the estimated p, or the estimated probability of respondents supporting the use of facial recognition technology.

```r
mean(dta_sel$yes_fr, na.rm=T)
```

```
[1] 0.234308
```

3.1.1 Summary

To sum up the whole process again, when we only include the intercept in the logit regression model:

- $\log\left(\frac{p}{1-p}\right) = \alpha$; the estimated intercept is the estimated **log odds**,
- $e^{\alpha} = \frac{p}{1-p}$ the exponent of the estimated intercept is the estimated **odds**, and
- once we solve the equation for p it follows that $p = \frac{e^{\alpha}}{1+e^{\alpha}}$, the estimated probability of $Y = 1$, where Y is the outcome variable of interest.

3.2 With Predictor(s)

Let's add predictors to the logit regression model. For simplicity, say we want to use `highedu` to explain the dependent variable.

With the predictor included, the log odds of a respondent supporting the police's use of facial recognition is

$$\log\left(\frac{p}{1-p}\right) = \alpha + \beta(\text{highedu}).$$

Let's run the analysis.

```r
mod_edu <- glm(yes_fr ~ highedu, data = dta_sel, family = binomial)
summary(mod_edu)
```

```
Call:
glm(formula = yes_fr ~ highedu, family = binomial, data = dta_sel)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.11163    0.03265 -34.051  < 2e-16 ***
highedu     -0.14520    0.04665  -3.113  0.00185 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 11172  on 10259  degrees of freedom
Residual deviance: 11162  on 10258  degrees of freedom
AIC: 11166
```

```
Number of Fisher Scoring iterations: 4
```

The estimated coefficient for `highedu` is -0.14520 and statistically significant with $p < 0.05$. How should we interpret the results?

In contrast to the model with no predictor, taking the exponent of the coefficient will give us the **odds ratio** (OR). OR is notoriously confusing, but here is the basic intuition with minimum math involved.

First, recall that we define the model as

$$\log\left(\frac{p}{1-p}\right) = \alpha + \beta(\text{highedu}) + \epsilon,$$

where the outcome variable is the log odds of Americans in favor of facial recognition technology.

Next, β tells us how the *log odds* of people supporting facial recognition technology will change when we increase the predictor from 0 to 1. Since our predictor `highedu` is a binary variable, we are basically comparing respondents who went to the colleges ($X = 1$) to those who did ($X = 0$):

$$\beta = \log(\text{Odds when X} = 1) - \log(\text{Odds when X} = 0),$$

where by "Odds" we mean **the odds of respondents supporting the use of facial recognition technology by the police**.

By **the quotient law of logarithms**, the **the difference between two logs** (the left hand side below) can be rewritten as **the log of them dividing up each other** (the right hand side below):

$$\beta = \log(\text{Odds when X} = 1) - \log(\text{Odds when X} = 0) = \log\left(\frac{\text{Odds when X} = 1}{\text{Odds when X} = 0}\right) \quad (1)$$

In other words, the estimated slope $\hat{\beta}$ is the log of our odds ratio with base e. Taking the exponent of $\hat{\beta}$ will return odds ratio. In doing so, we have

$$e^{\beta} = \frac{\text{Odds when X} = 1}{\text{Odds when X} = 0}.$$

This is **the odds ratio of the support for facial recognition when respondents have a university degree or above.**

Let's do this step by step. Taking the exponent of our $\hat{\beta}$ will return a value below 1, suggesting the odds of supporting secession when $X = 1$ is lower than the odds of supporting secession when $X = 0$ (i.e., the denominator is larger than the numerator).

```r
beta <- coef(mod_edu)
exp(beta)
```

```
(Intercept)     highedu 
  0.3290237   0.8648459 
```

We can use `confint()` to get the confidence interval of the odds ratio of `highedu` to see if it is stays below 1.

```r
coefs_ci <- confint(mod_edu) # CIs of estimated beta
coefs_ci
```

```
                 2.5 %      97.5 %
(Intercept) -1.1759618 -1.04797905
highedu     -0.2366728 -0.05380172
```

```r
exp(coefs_ci) # CIs of odds ratio
```

```
                2.5 %    97.5 %
(Intercept) 0.3085221 0.3506457
highedu     0.7892495 0.9476200
```

In a nutshell, odds-ratio (OR) is **a ratio of two different odds** when we change the predictor `highedu` by one unit. And OR can only be one of the three conditions (see the table below), each of which has a different substantive implication.

Since we find $OR_{highedu} < 1$ even after we consider its confidence interval, people with a university degree are **less** likely to support secession.

3.2.1 Summary

Again, we can sum up the process as follows when we include predictor(s) in the logit regression model:

- $\beta = \log\left(\frac{\text{Odds when X=1}}{\text{Odds when X=0}}\right)$; the estimated slope is the estimated **log of odds ratio**,
- $e^{\beta} = \frac{\text{Odds when X=1}}{\text{Odds when X=0}}$ the exponent of the estimated slope is the estimated **odds ratio**
- You can still uncover p when you include predictors, and $p = \frac{e^{\alpha+\beta X}}{1+e^{\alpha+\beta X}}$.

	It means	So when respondents have a uni degree	Put it simply
OR=1	(Odds when X=1) = (Odds when X=0)	the odds/chance of supporting AI remains the same	Unclear how education is correlated with attitudes toward AI
OR>1	(Odds when X=1) > (Odds when X=0)	the odds/chance of supporting AI is higher	Highly educated respondents are more likely to support AI
OR<1	(Odds when X=1) < (Odds when X=0)	the odds/chance of supporting AI is lower	Highly educated respondents are less likely to support AI

3.3 Practical Guide for Logit Regression

When you conduct a logit regression (with predictors), do the following:

- Step 1: Write down a table like the one above to think through different scenarios.
- Step 2: Run the logit regression (use `glm()`); see if your predictor(s) have any statistically significant coefficients.
- Step 3: Take the exponent of your coefficients (use `exp()`); this will be the odds ratio when we change the corresponding predictor by one unit.
- Step 4: Explain the odds-ratio (OR) using the table you made.
  - If $OR = 1$, the predictor is not associated with the (odds of the) outcome (this scenario is going to be rare in practice).
  - If $OR > 1$, the predictor is positively associated with the (odds of the) outcome.
  - If $OR < 1$, the predictor is negatively associated with the (odds of the) outcome.
- Step 5: Obtain the confidence interval of the odds ratio to make sure it stays in one of the three situations.
- Step 6: Calculate the predicted probability (see below).

Problem Set

Question 1

Use `?glm` to see more information about `glm()`. What does the argument `family` do? What is the default for `family`?

Quesrion 2

Replicate the same analysis in Section 3.2, but now let's keep the `NA`s. In other words, recode `yes_fr` as follows instead:

```
dta_sel$yes_fr <- ifelse(dta_sel$FACEREC2_W99 == 1, 1, 0)
summary(dta_sel$yes_fr)
```

Can we still derive the same conclusion?

Question 3

Replace `highedu` with any variable in the same dataset and fit another bivariate logit regression. Answer the following questions.

- Interpret the estimated odds ratio with 95% confidence intervals.
- Compare the estimated odds ratio of `highedu` and the predictor you choose. Discuss your observations – can you say which one predictor has a larger influnece over the outcome variable than the other?
- Read through the codebook provided by Beesley and Cooper and find one variable and add it to the specified model. Present the table with `stargazer()` or `modelsummary()` function and discuss your observations.

4 Extra: Derive Predicted Probability from Logit Regression

Instead of using odds ratio, the most intuitive way to use the **`predict()`** function.

We can plug the model we have just estimated into `predict()` to compute the predicted **log-odds** for each observation. Including `type="response"` will return the predicted **probabilities**.

```r
fit_log_odds <- predict(mod_edu) # for predicted log odds
fit_prob <- predict(mod_edu, type="response") # for predicted probabilities
```

We can use the **`predict()`** function to estimate how the probability of respondents supporting secession will change depending on their language use at home.

First, let's use `data.frame()` to create a new dataframe to include the unique values of `highedu`.

```r
data_predict <- data.frame(highedu = c(0,1))
data_predict
```

```
  highedu
1       0
2       1
```

Now let's obtain the predicted log-odds, odds, and probabilities when we vary the value of `highedu`.

```r
fit_prob <- predict(mod_edu, newdata=data_predict, type="response")
fit_log_odds <- predict(mod_edu, newdata=data_predict)
fit_mod_s <- data.frame(highedu = c(0,1),
                        fit_prob = as.matrix(fit_prob),
                        fit_log_odds = as.matrix(fit_log_odds))
fit_mod_s$fit_odds <- exp(fit_mod_s$fit_log_odds)
fit_mod_s
```

```
  highedu  fit_prob fit_log_odds  fit_odds
1       0 0.2475680    -1.111625 0.3290237
2       1 0.2215202    -1.256829 0.2845548
```

We can see that the predicted probabilities of supporting secession for `highedu`= 0 and = 1 are 0.25 and 0.22, respectively. Again, highly educated respondents are **less** likely to support AI.

5 Extra: Multiple Logit Regression

```r
dta_ext <- dta_sel |>
  dplyr::select(yes_fr, female, democrat, highedu) |>
  drop_na() # remove the NAs for model comparison
```

5.1 Fit the Model

```r
mod1 <- glm(yes_fr ~ highedu,
            data = dta_ext, family = binomial)
mod2 <- glm(yes_fr ~ highedu + democrat,
            data = dta_ext, family = binomial)
mod3 <- glm(yes_fr ~ highedu + democrat + female,
            data = dta_ext, family = binomial)
```

5.2 Regression Table

```r
library(stargazer)
stargazer(list(mod1, mod2, mod3),
          omit.stat = c("f", "rsq", "ser"),
          covariate.labels = c("University graduate (=1)",
                               "Democrat (=1)",
                               "Female (=1)"),
          type = "text",
          digits = 3,
          no.space = T,
          intercept.bottom = TRUE,
          star.cutoffs = c(0.05, 0.01, 0.001))
```

```
=========================================================
                               Dependent variable:
                         --------------------------------
                                      yes_fr
                            (1)         (2)         (3)
---------------------------------------------------------
University graduate (=1)  -0.145**    -0.138**    -0.136**
                          (0.047)     (0.047)     (0.047)
```

```
Democrat (=1)                          -0.112*    -0.115*
                                       (0.051)    (0.051)
Female (=1)                                        0.024
                                                  (0.047)
Constant                    -1.112***  -1.080***  -1.093***
                             (0.033)    (0.036)    (0.044)
---------------------------------------------------------
Observations                  10,260     10,260     10,260
Log Likelihood              -5,580.999 -5,578.529 -5,578.405
Akaike Inf. Crit.           11,166.000 11,163.060 11,164.810
=========================================================
Note:                          *p<0.05; **p<0.01; ***p<0.001
```

5.3 Model Comparison

```
anova(mod1, mod2, mod3, test="Chi")
```

```
Analysis of Deviance Table

Model 1: yes_fr ~ highedu
Model 2: yes_fr ~ highedu + democrat
Model 3: yes_fr ~ highedu + democrat + female
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     10258      11162
2     10257      11157  1   4.9399  0.02624 *
3     10256      11157  1   0.2480  0.61852
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.4 Sum of Squared Residuals

```
sum(residuals(mod1)^2)
```

```
[1] 11162
```

```
sum(residuals(mod2)^2)
```

```
[1] 11157.06
```

```r
sum(residuals(mod3)^2)
```

```
[1] 11156.81
```

5.5 Marginal Predicted Probability

We can obtain the predicted probabilities of education using the same function `predict()`. We will again recreate a new dataframe to record the unique values of `highedu`.

As we will be using a multiple logit model `mod3` to carry out the simulation/prediction exercise, we will have to set other variables in the model at its mean.

```r
data_predict <- data.frame(highedu = c(0,1),
                           democrat = mean(dta_ext$democrat, na.rm=T),
                           female = mean(dta_ext$female, na.rm=T))
data_predict
```

```
  highedu  democrat   female
1       0 0.3246589 0.548538
2       1 0.3246589 0.548538
```

The rest follows the same process as above.

```r
fit_prob <- predict(mod3, newdata=data_predict, type="response")
fit_log_odds <- predict(mod3, newdata=data_predict)
fit_mod_s <- data.frame(highedu = c(0,1),
                        fit_prob = as.matrix(fit_prob),
                        fit_log_odds = as.matrix(fit_log_odds))
fit_mod_s$fit_odds <- exp(fit_mod_s$fit_log_odds)
fit_mod_s
```

```
  highedu  fit_prob fit_log_odds  fit_odds
1       0 0.2465719    -1.116980 0.3272667
2       1 0.2221527    -1.253165 0.2855994
```

6 Extra: Including Weights in the Analysis

The `survey` and `srvyr` packages makes it easy to account for survey weights in our analysis.

The function `svyglm()` is similar to the conventional `glm()` function except that `svyglm()` use `design` (i.e., the survey object) instead of `data`.

Type `?svyglm` for more information about the function. Alternatively, you can visit the `survey` package's vignette page. The section **Regression models** provides additional information.

In this section, we will use the updated data frame `dta_sel`, which should have included all the variables we have added so far, to carry out the following activities.

- Use `as_survey` from `srvyr` to create a survey design object.
- Use `svyglm()` from `survey` to fit logit regression.

In general, including the weights should not impact the results much unless the sample is very biased in relation to the (pre-specified) population.

```r
dta_sel <- dta %>%
  dplyr::select(QKEY,
                FACEREC2_W99,
                F_GENDER, F_EDUCCAT, F_PARTY_FINAL,
                WEIGHT_W99) %>%
  mutate(yes_fr = if_else(FACEREC2_W99 == 1, 1, 0),
  highedu = if_else(F_EDUCCAT == 1, 1, 0),
  female = if_else(F_GENDER == 2, 1, 0),
  democrat = if_else(dta_sel$F_PARTY_FINAL == 2, 1, 0)) %>%
  unique()
```

6.1 Create Survey Design Object

Note that the original dataset does not include much information about the sampling scheme, so it is not clear how the respondents are sampled.

Below we assume the Pew Research Center created the study population (i.e., the sample) with stratification by oblast (use the `oblastname` variable). Since there is no information about the population size of each oblast, we will ignore it.

```
dta_sel_survey <- dta_sel %>%
  srvyr::as_survey(ids = QKEY,
             weights = WEIGHT_W99)
dta_sel_survey
```

```
Independent Sampling design (with replacement)
Called via srvyr
Sampling variables:
 - ids: QKEY
 - weights: WEIGHT_W99
Data variables: QKEY (dbl), FACEREC2_W99 (dbl+lbl), F_GENDER (dbl+lbl),
  F_EDUCCAT (dbl+lbl), F_PARTY_FINAL (dbl+lbl), WEIGHT_W99 (dbl), yes_fr (dbl),
  highedu (dbl), female (dbl), democrat (dbl)
```

6.2 Fit Logit Regression

To carry out logit regression with weights included, we need to use `svyglm()` in the `survey` package.

```
mod_s_edu <- survey::svyglm(yes_fr ~ highedu,
                              design = dta_sel_survey, # the survey design object
                              family = binomial)
```

```
Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(mod_s_edu)
```

```

Call:
svyglm(formula = yes_fr ~ highedu, design = dta_sel_survey, family = binomial)

Survey design:
Called via srvyr

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.07663    0.06335  -1.210 0.226460
highedu     -0.30798    0.08359  -3.684 0.000232 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 0.9941408)

Number of Fisher Scoring iterations: 3
```

Let's take a look at the odds-ratio by taking the exponent of the `highedu` coefficient.

```
exp(coef(mod_s_edu))
```

```
(Intercept)     highedu
  0.9262354   0.7349275
```