Chapter 13: Generalised Linear Models

Answers to Exercises

Brian Fogarty

Contents

Exercise 1 - Logit Model

Exercise 1.a

Before running the logit regression, we need to do some data wrangling:

- Flip `brexit_vote` so "2. Leave" is the higher category.
- Correctly order `vfproblem` and then convert it to a numeric variable.
- Re-order `pid` so "Other" is the comparison category.

```
library(tidyverse)

vf_england <- read_csv("VF England.csv")

vf_england <- vf_england %>%
  mutate(brexit_vote1 = fct_rev(as_factor(brexit_vote)),
         vfproblem1 = as.numeric(factor(vfproblem,
                        levels = c("Strongly disagree","Disagree",
```

```
                          "Slightly disagree","Neither agree nor disagree",
                          "Slightly agree","Agree","Strongly agree"))),
         pid1 = factor(pid,
                  levels = c("Other","Conservative","UKIP Brexit"))
         )


vf_england %>%
  count(brexit_vote1)
```

```
# A tibble: 3 x 2
  brexit_vote1     n
  <fct>        <int>
1 1. Remain      864
2 2. Leave       875
3 <NA>           295
```

```
vf_england %>%
  count(vfproblem1)
```

```
# A tibble: 7 x 2
  vfproblem1     n
       <dbl> <int>
1          1   112
2          2   245
3          3   302
4          4   670
5          5   398
6          6   167
7          7   140
```

```
vf_england %>%
  count(pid1)
```

```
# A tibble: 3 x 2
  pid1             n
  <fct>        <int>
1 Other         1351
2 Conservative   550
3 UKIP Brexit    133
```

Exercise 1.b

```
summary(model.logit <- glm(brexit_vote1 ~ vfproblem1 + pid1 + age,
                           family = binomial(link = "logit"), data = vf_england))
```

```
Call:
glm(formula = brexit_vote1 ~ vfproblem1 + pid1 + age, family = binomial(link = "logit"),
    data = vf_england)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-3.02718  -0.93490   0.08521   1.02232   1.85371
```

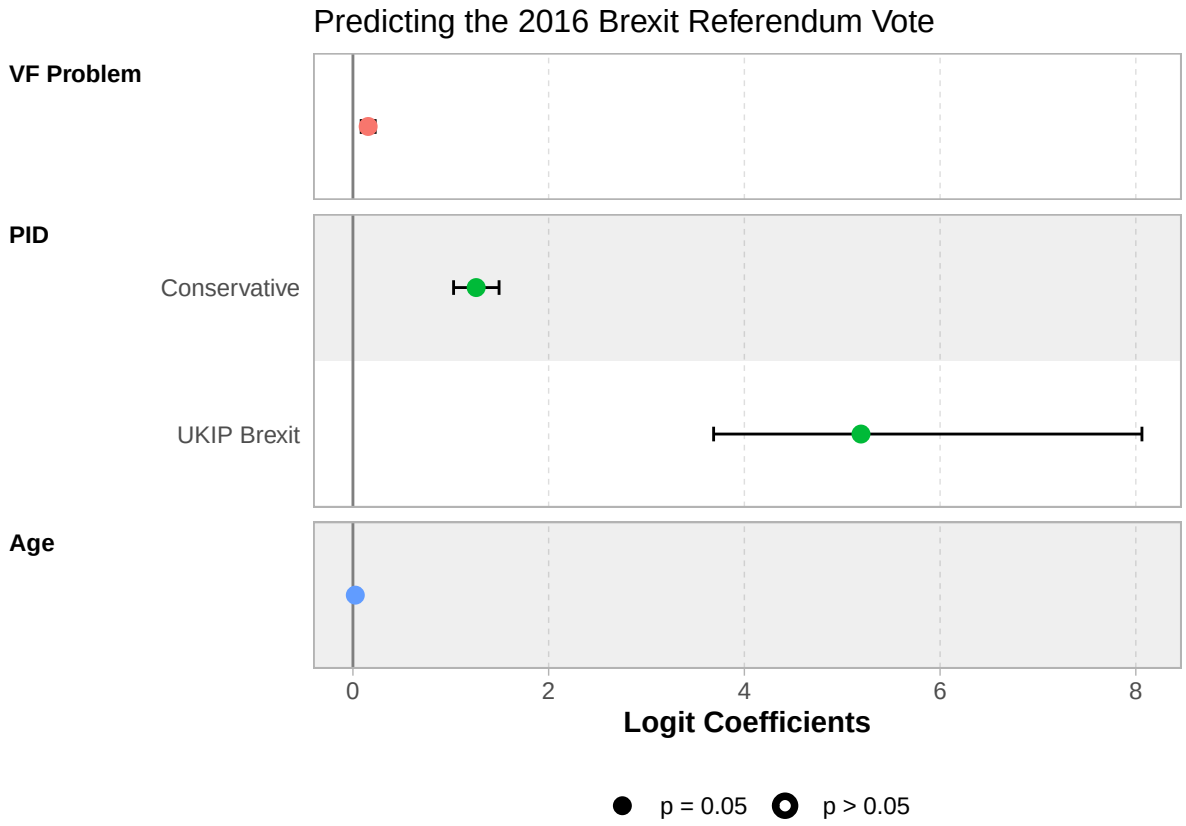
```
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -2.415150   0.231850 -10.417  < 2e-16 ***
vfproblem1        0.155748   0.036423   4.276 1.90e-05 ***
pid1Conservative  1.259450   0.118440  10.634  < 2e-16 ***
pid1UKIP Brexit   5.191752   1.007051   5.155 2.53e-07 ***
age               0.024303   0.003536   6.873 6.28e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2410.7  on 1738  degrees of freedom
Residual deviance: 1999.5  on 1734  degrees of freedom
  (295 observations deleted due to missingness)
AIC: 2009.5

Number of Fisher Scoring iterations: 7
```

We see that all of the predictors have a positive and statistically significant effect on Brexit vote.

Exercise 1.c

We'll use the `ggcoef_model()` function from the `GGally` package for the coefficient plot. We'll include the option `no_reference_row = "pid1"` to remove the "Other" category from the plot.

```
library(GGally)

ggcoef_model(model.logit,
             variable_labels = c(
               vfproblem1 = "VF Problem",
               pid1 = "PID",
               age = "Age"),
             no_reference_row = "pid1",
             show_p_values = FALSE,
             signif_stars = FALSE) +
  labs(title = "Predicting the 2016 Brexit Referendum Vote",
       x = "Logit Coefficients") +
  theme(
    plot.title = element_text(size = 12)
  )
```



Exercise 1.d

```
(exp(model.logit$coefficients[-1])-1)*100
```

```
      vfproblem1 pid1Conservative  pid1UKIP Brexit             age
       16.853212       252.348284     17878.318620        2.460048
```

For a one-unit increase in the belief that voter fraud is a problem, the odds of voting "Leave" increase by 16.85%.

For Conservative Party identifiers, the odds of voting "Leave" are 252.35% greater than identifiers of "other" parties. (Specifying '"other" parties' is a bit awkward here, so we could instead say *identifiers of other parties (excluding UKIP/Brexit Party identifiers)* to make it clearer; though it is still awkward.)

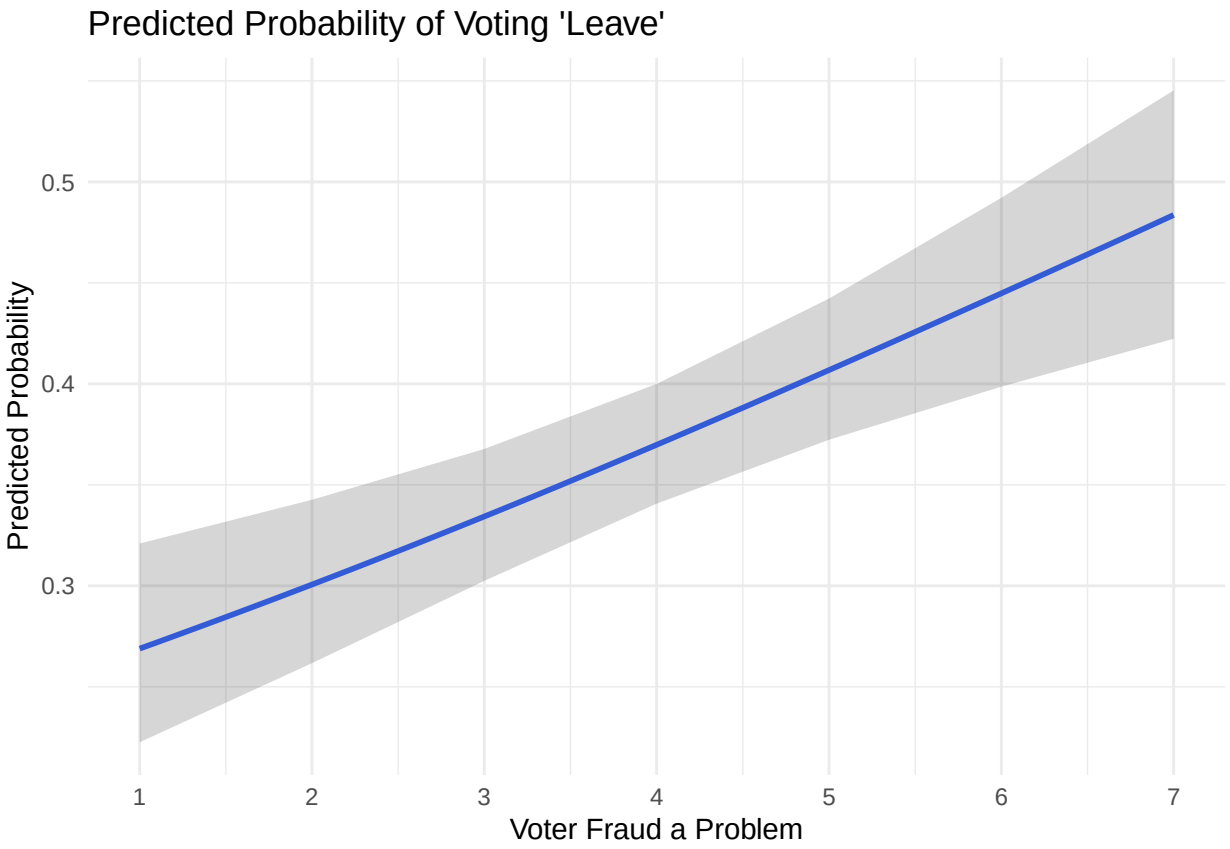
For UKIP/Brexit Party identifiers, the odds of voting "Leave" are 17,878.32% greater than identifiers of "other" parties. (Not only do we have the awkward '"other" parties' phrasing, but the odds ratio value is absurd. However, and obviously, the number one issue for the Brexit Party was to get the UK to leave the EU. Therefore, we should expect an absurd value.)

For a one-unit increase in age, the odds of voting "Leave" increase by 2.46%.

Exercise 1.e

We'll use the `ggpredict()` function from the `ggeffects` package to plot the predicted probabilities for `vfproblem1`.

```
library(ggeffects)

ggpredict(model.logit, terms = "vfproblem1") %>%
ggplot(mapping = aes(x = x, y = predicted)) +
  geom_smooth(se = FALSE) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .2) +
  scale_x_continuous(limits = c(1,7),breaks = c(1:7)) +
  labs(title = "Predicted Probability of Voting 'Leave'",
       x = "Voter Fraud a Problem", y = "Predicted Probability") +
  theme_minimal()
```



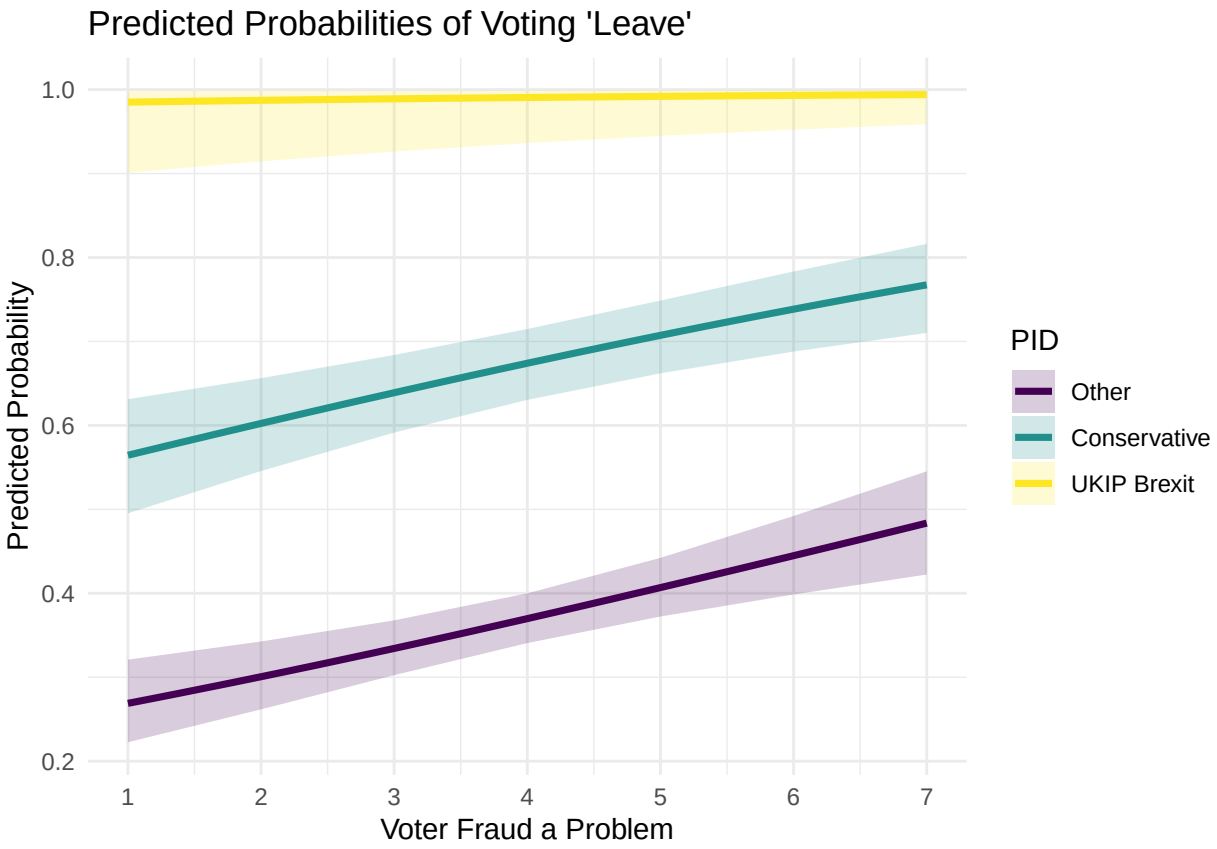
We see that *as the belief that voter fraud is a problem increases, the predicted probability of voting "Leave" increases (in a nearly linear manner).* (It is somewhat uncommon for predicted probabilities to be linear). If we wanted to add specifics, we could say something like *there is roughly a .2 (20%) increase in the predicted probability of voting "Leave" between the lowest belief ("Strongly disagree") and the highest belief ("Strong agree") that voter fraud is a problem.*

Exercise 1.f

Our `pid1` variable already has labels and so we don't have to re-do them in the code.

```
ggpredict(model.logit, terms = c("vfproblem1","pid1")) %>%
ggplot(mapping=aes(x = x, y = predicted, colour = group, fill = group)) +
  geom_smooth(se = FALSE, size = 1.25) +
```

```
geom_ribbon(aes(ymin = conf.low, ymax = conf.high),
                alpha = .2, colour = NA) +
scale_x_continuous(limits = c(1,7), breaks = c(1:7)) +
labs(title = "Predicted Probabilities of Voting 'Leave'",
      x = "Voter Fraud a Problem", y = "Predicted Probability") +
guides(colour = guide_legend(title = "PID"),
       fill = guide_legend(title = "PID")) +
theme_minimal() +
scale_fill_viridis_d() +
scale_colour_viridis_d()
```



We may interpret and discuss this plot as the following:

> *For all party identifiers, as the belief that voter fraud is problem increases, the predicted probability of voting "Leave" increases. We also see that UKIP/Brexit Party identifiers have the highest predicted probability of voting "Leave" and this probability barely increases as beliefs that voter fraud is a problem increases. Conservative Party and identifiers of other parties appear to have a similar increase in the predicted probability of voting "Leave" as beliefs that voter fraud is a problem increases.*

Exercise 2 - Ordered Logit Model

Exercise 2.a

We will use two lines to recode `vfproblem`. First, we'll collapse the "disagree" and "agree" categories using the `fct_collapse()` function. Second, we'll re-order the levels and specify that the variable is ordered.

```
vf_england <- vf_england %>%
  mutate(vfproblem2 = fct_collapse(vfproblem,
                         "Disagree" = c("Strongly disagree","Disagree","Slightly disagree"),
                         "Agree" = c("Strongly agree","Agree","Slightly agree")),
         vfproblem2 = ordered(factor(vfproblem2,
                          levels = c("Disagree","Neither agree nor disagree","Agree")))
         )

vf_england %>%
  count(vfproblem2)
```

```
# A tibble: 3 x 2
  vfproblem2                     n
  <ord>                      <int>
1 Disagree                     659
2 Neither agree nor disagree   670
3 Agree                        705
```

Exercise 2.b

```
library(MASS)

summary(model.ologit <- polr(vfproblem2 ~ brexit_vote1 + pid1 + age,
                             method = "logistic", data = vf_england))
```

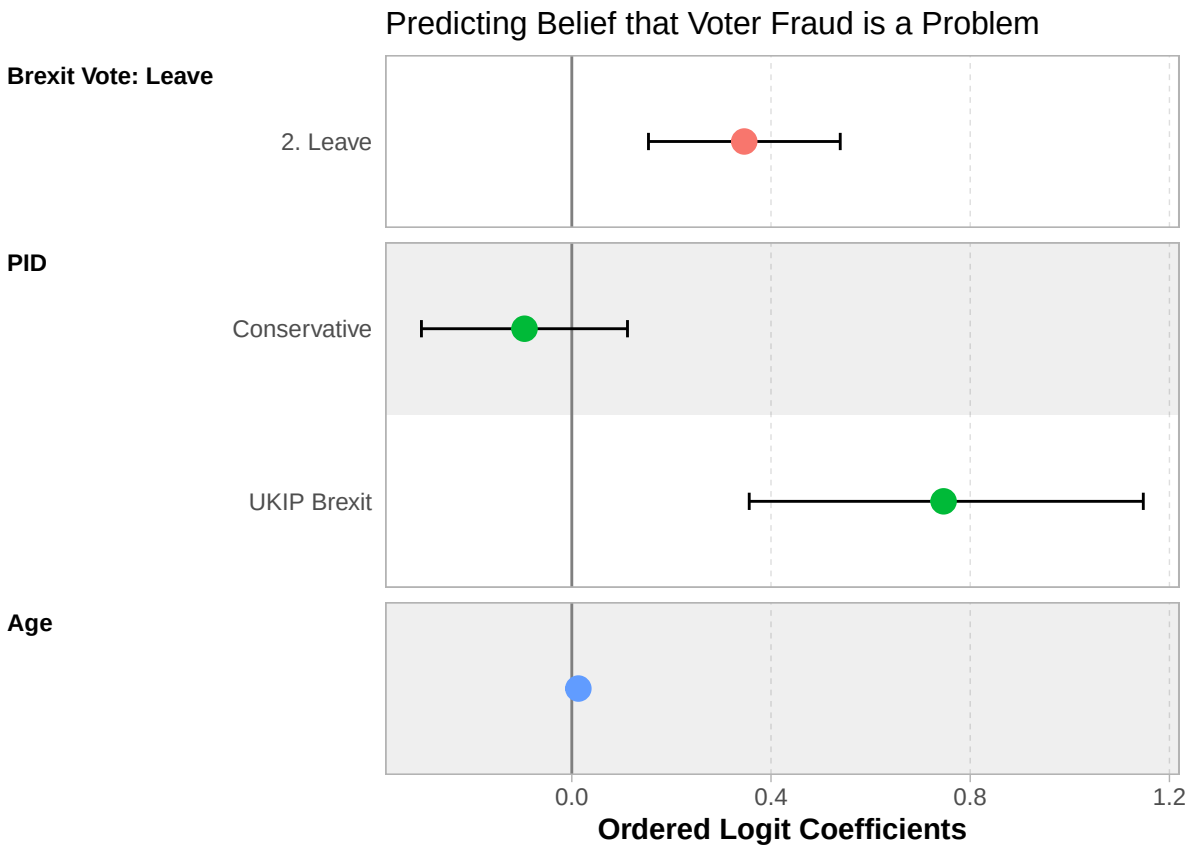
```
Call:
polr(formula = vfproblem2 ~ brexit_vote1 + pid1 + age, data = vf_england,
    method = "logistic")

Coefficients:
                        Value Std. Error t value
brexit_vote12. Leave  0.34630   0.098204  3.5263
pid1Conservative     -0.09490   0.105514 -0.8994
pid1UKIP Brexit       0.74655   0.201507  3.7048
age                   0.01319   0.002993  4.4069

Intercepts:
                                    Value   Std. Error t value
Disagree|Neither agree nor disagree  0.1290  0.1526     0.8455
Neither agree nor disagree|Agree     1.4544  0.1568     9.2745

Residual Deviance: 3739.316
AIC: 3751.316
(295 observations deleted due to missingness)
```

Based on the t-values, we see that `brexit_vote`, UKIP/Brexit Party identifiers (from `pid1`), and `age` have a positive and statistically significant effect on `vfproblem2`.

Exercise 2.c

We'll again use the `ggcoef_model()` function from the `GGally` package for the coefficient plot. We'll include the option `no_reference_row = c("brexit_vote1","pid1")` to remove the comparison categories from the plot.

```
ggcoef_model(model.ologit,
               variable_labels = c(
                 brexit_vote1 = "Brexit Vote: Leave",
                 pid1 = "PID",
                 age = "Age"),
               no_reference_row = c("brexit_vote1","pid1"),
               show_p_values = FALSE,
               signif_stars = FALSE) +
  labs(title = "Predicting Belief that Voter Fraud is a Problem",
       x = "Ordered Logit Coefficients") +
  theme(
    plot.title = element_text(size = 12)
  )
```



Exercise 2.d

We'll use the `brant()` function from the `brant` package to test the parallel regression assumption.

```
library(brant)
brant(model.ologit)
```

```
------------------------------------------------------
Test for        X2  df  probability
------------------------------------------------------
Omnibus         30.17   4   0
brexit_vote12. Leave    0.33    1   0.57
pid1Conservative    2.51    1   0.11
pid1UKIP Brexit 12.49   1   0
age         10.78   1   0
------------------------------------------------------

H0: Parallel Regression Assumption holds
```

```
Warning in brant(model.ologit): 2 combinations in table(dv,ivs) do not occur.
Because of that, the test results might be invalid.
```

The warning message tells us that two of the possible combinations of values between our outcome variable and predictors have no observations. The weird values for "UKIP Brexit" (in `pid1`) suggest it's the culprit. Let's check:

```
vf_england %>%
  group_by(vfproblem2,brexit_vote1,pid1) %>%
  filter(pid1 == "UKIP Brexit") %>%
  count(vfproblem2)
```

```
# A tibble: 7 x 4
# Groups:   vfproblem2, brexit_vote1, pid1 [7]
  vfproblem2                 brexit_vote1 pid1            n
  <ord>                      <fct>        <fct>       <int>
1 Disagree                   2. Leave     UKIP Brexit    28
2 Disagree                   <NA>         UKIP Brexit     2
3 Neither agree nor disagree 2. Leave     UKIP Brexit    19
4 Neither agree nor disagree <NA>         UKIP Brexit     2
5 Agree                      1. Remain    UKIP Brexit     1
6 Agree                      2. Leave     UKIP Brexit    80
7 Agree                      <NA>         UKIP Brexit     1
```

Yep, there are no UKIP/Brexit Party identifiers for "Disagree" (on `vfproblem2`) and "1. Remain" (on `brexit_vote1`), and "Neither agree nor disagree" (on `vfproblem2`) and "1. Remain" (on `brexit_vote1`).

The warning message says the test might be invalid; in other words, we can't trust the results. Currently, the `Omnibus` tells us that we violate PRA, and even without the problem in the data we might still violate PRA. But, it is reasonable to argue that our outcome variable is ordered and thus we should use an ordered outcome regression model (i.e., ordered logit).

Exercise 2.e

```
(exp(model.ologit$coefficients)-1)*100
```

```
brexit_vote12. Leave     pid1Conservative      pid1UKIP Brexit
           41.382346            -9.053954           110.970346
```

```
     age
1.327523
```

We don't interpret the effect for "Conservative" since it's not statistically significant.

For "Leave" voters, the odds of having a higher belief that voter fraud is a problem are 41.38% greater than "Remain" voters.

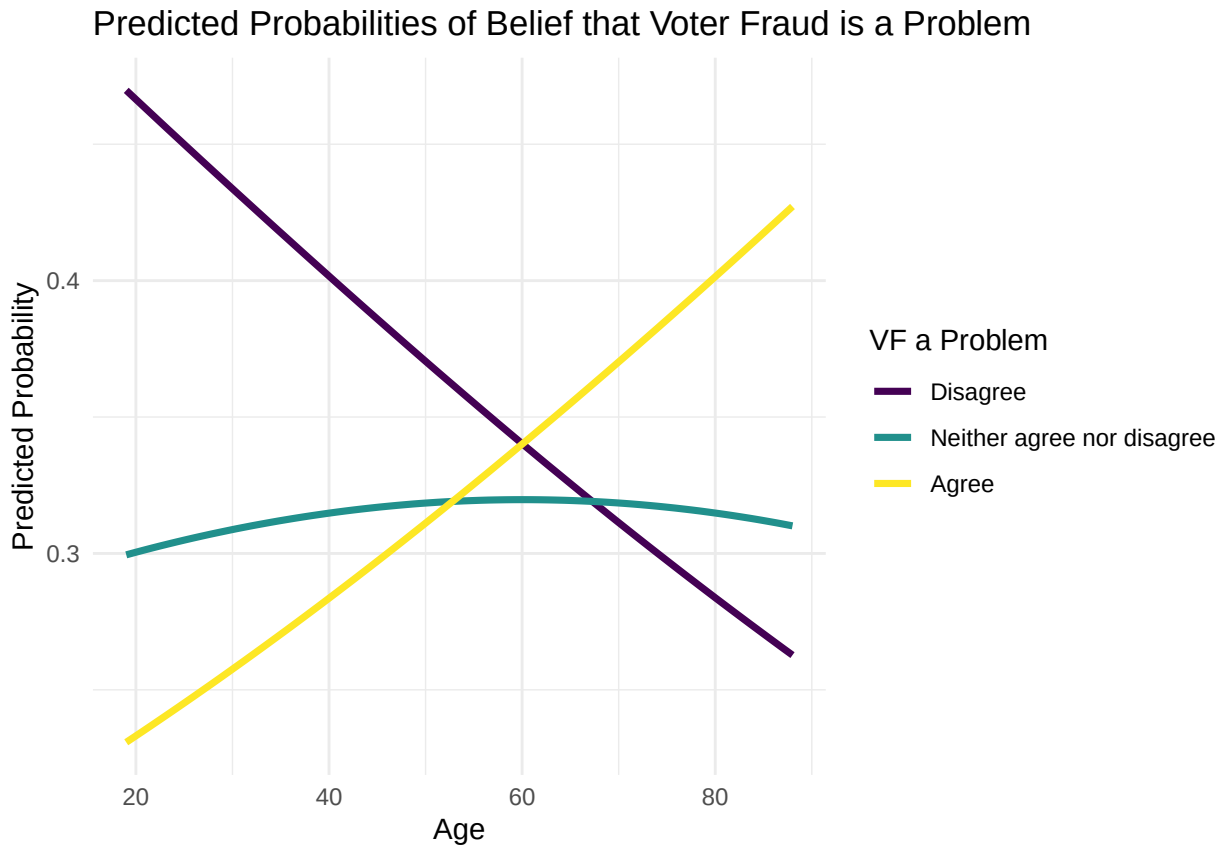
For UKIP/Brexit Party identifiers, the odds of having a higher belief that voter fraud is a problem are 110.97% greater than identifiers of "other" parties.

For a one-unit increase in age, the odds of having a higher belief that voter fraud is a problem increase by 1.33%.

Exercise 2.f

We'll use `age` on the x-axis of the plot since it's the only non-nominal predictor. We'll include `terms = "age [all]"` in the `ggpredict()` function; the function actually will tell us to add `[all]` to create a nicer appearance if we just specify `terms = "age"`. We'll also cut the prediction intervals because it makes the plot harder to understand.

```
ggpredict(model.ologit, terms = "age [all]") %>%
  mutate(response.level = ordered(as_factor(response.level))) %>%
ggplot(mapping=aes(x = x, y = predicted, colour = response.level,
                   fill = response.level)) +
  geom_smooth(se = FALSE, size = 1.25) +
  labs(title = "Predicted Probabilities of Belief that Voter Fraud is a Problem",
       x = "Age", y = "Predicted Probability") +
  guides(colour = guide_legend(title = "VF a Problem"),
         fill = guide_legend(title = "VF a Problem")) +
  theme_minimal() +
  scale_fill_viridis_d() +
  scale_colour_viridis_d()
```



Broadly, we can interpret the plot with the following:

> *We see that as age increases, the predicted probability of disagreeing that voter fraud is a problem decreases, neither agreeing nor disagreeing slightly increases and then slightly decreases, and agreeing increases. We see that disagree has the highest predicted probability for young through middle aged respondents. For respondents who are 60 or older, agree has the highest predicted probability.*

Exercise 3 - Multinomial Logit Model

Exercise 3.a

Although partisan identification is not commonly used as an outcome variable, we'll use it here for the sake of the exercise.

```
library(nnet)
summary(model.mlogit <- multinom(pid1 ~ brexit_vote1 + vfproblem1 + age,
                                 data = vf_england))
```

```
# weights:  15 (8 variable)
initial  value 1910.486770
iter  10 value 1333.987887
iter  20 value 1255.580048
final  value 1255.576652
converged
```

```
Call:
multinom(formula = pid1 ~ brexit_vote1 + vfproblem1 + age, data = vf_england)

Coefficients:
             (Intercept) brexit_vote12. Leave  vfproblem1        age
Conservative   -2.819536             1.262019 -0.04887307 0.03024652
UKIP Brexit    -8.884808             5.173903  0.27978666 0.02384952

Std. Errors:
             (Intercept) brexit_vote12. Leave vfproblem1         age
Conservative   0.2536372            0.1182306 0.03752915 0.003937031
UKIP Brexit    1.1024427            1.0070598 0.06794136 0.007482982

Residual Deviance: 2511.153
AIC: 2527.153
```

```
summary(model.mlogit)$coefficients/summary(model.mlogit)$standard.errors
```

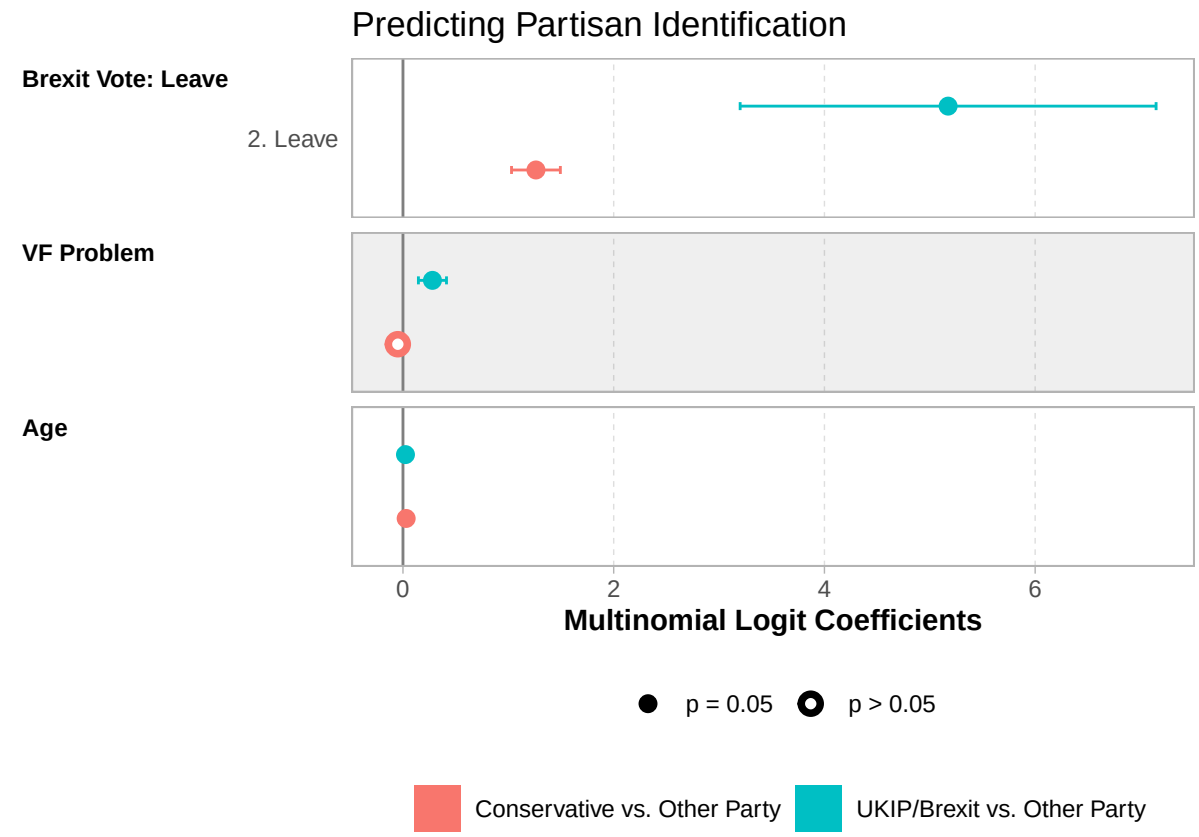
```
             (Intercept) brexit_vote12. Leave vfproblem1      age
Conservative  -11.116415            10.674215  -1.302269 7.682571
UKIP Brexit    -8.059202             5.137633   4.118061 3.187167
```

We find that all three predictors are statistically significant predictors of both comparisons in the outcome variable, except for "Conservative" and `vfproblem1`.

Exercise 3.b

```
ggcoef_multinom(model.mlogit,
                variable_labels = c(
                  brexit_vote1 = "Brexit Vote: Leave",
                  vfproblem1 = "VF Problem",
                  age = "Age"),
                show_p_values = FALSE,
                signif_stars = FALSE,
                no_reference_row = c("brexit_vote1"),
                y.level_label = c(
```

```
                  "Conservative" = "Conservative vs. Other Party",
                  "UKIP Brexit" = "UKIP/Brexit vs. Other Party"
                )) +
  labs(title = "Predicting Partisan Identification",
       x = "Multinomial Logit Coefficients")
```



Exercise 3.c

```
(exp(coef(model.mlogit))-1)*100
```

```
             (Intercept) brexit_vote12. Leave vfproblem1      age
Conservative   -94.03664             253.2546   -4.76980 3.070859
UKIP Brexit    -99.98615           17560.2800   32.28476 2.413619
```

For "Leave" voters, the odds of identifying with the Conservative Party compared to another party (excluding the UKIP/Brexit Party) are 253.25% greater than "Remain" voters. (Yeah, this is a mouthful.)

For "Leave" voters, the odds of identifying with the UKIP/Brexit Party compared to another party (excluding the Conservative Party) are 17,560.28% greater than "Remain" voters. (Yes, this is an absurd value.)

For a one-unit increase in the belief that voter fraud is a problem, the odds of identifying with the UKIP/Brexit Party compared to another party (excluding the Conservative Party) increase by 32.28%.

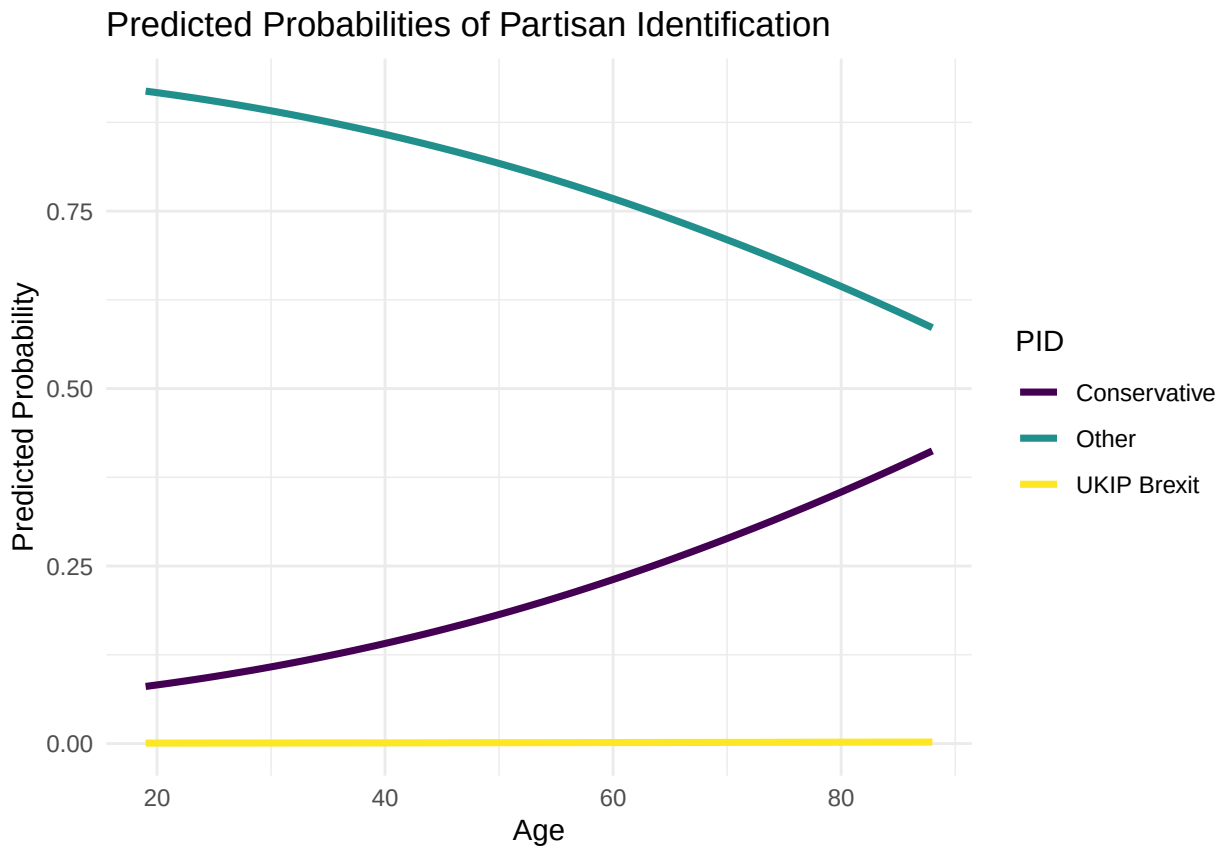
For a one-unit increase in age, the odds of identifying with the Conservative Party compared to another party (excluding the UKIP/Brexit Party) increase by 3.07%.

For a one-unit increase in age, the odds of identifying with the UKIP/Brexit Party compared to another party (excluding the Conservative Party) increase by 2.41%.

Exercise 3.d

Because `vfproblem1` is not statistically significant for both comparisons, we should not use it in our predicted probability plot. Instead, we'll use `age`.

```
ggpredict(model.mlogit, terms = "age [all]") %>%
ggplot(mapping = aes(x = x, y = predicted, colour = response.level)) +
  geom_smooth(se = FALSE, size = 1.25) +
  labs(title = "Predicted Probabilities of Partisan Identification",
       x = "Age", y = "Predicted Probability") +
  guides(colour = guide_legend(title = "PID")) +
  theme_minimal() +
  scale_fill_viridis_d() +
  scale_colour_viridis_d()
```



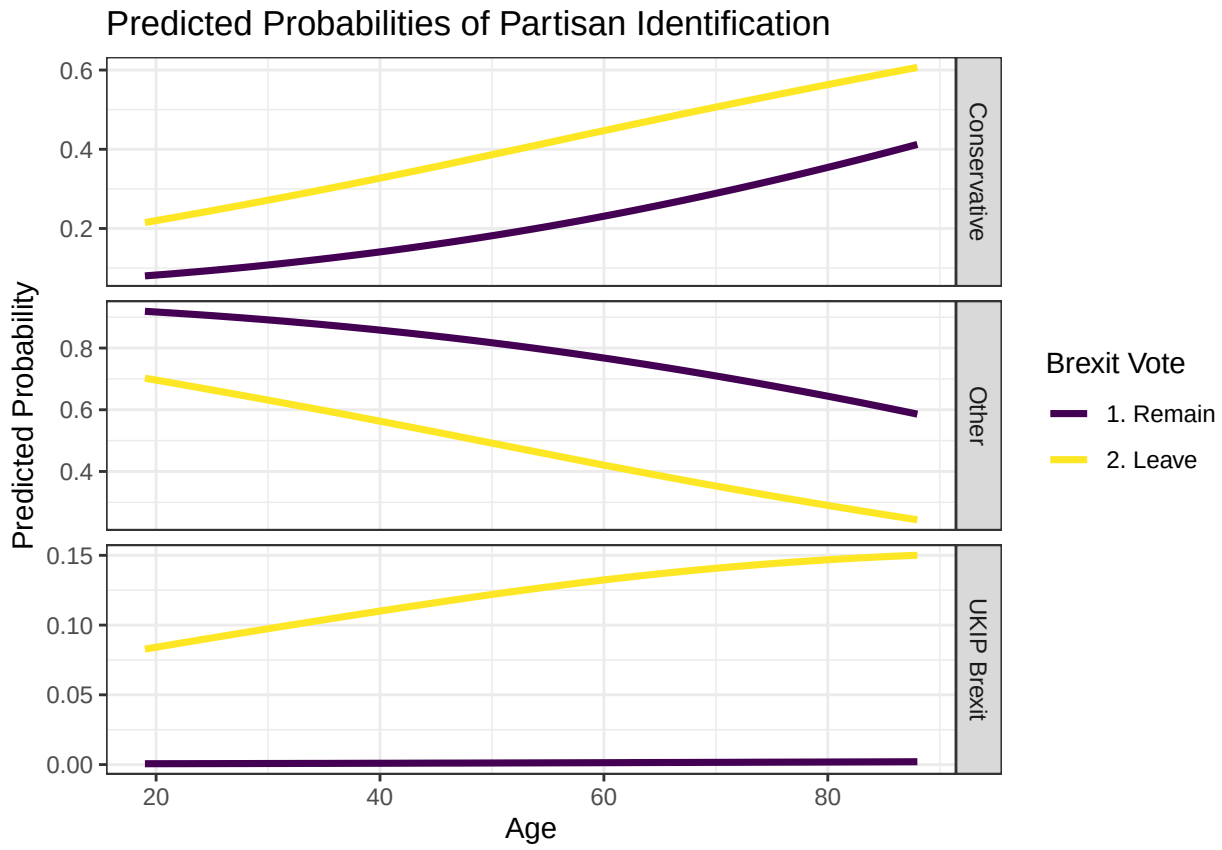
Broadly, we can interpret the plot with the following:

> *We see that as age increases, the predicted probability of identifying with the Conservative Party increases, identifying with the UKIP/Brexit Party is roughly flat, and identifying with a different party decreases. Essentially, as people get older they are more likely to identify with the Conservative Party; though there may be cohort effects we are not observing.*

Exercise 3.e

Now, we'll add `brexit1` to the plot.

```
ggpredict(model.mlogit, terms = c("age [all]","brexit_vote1")) %>%
ggplot(mapping = aes(x = x, y = predicted, colour = group, fill = group)) +
  geom_smooth(se = FALSE, size = 1.25) +
  labs(title = "Predicted Probabilities of Partisan Identification",
       x = "Age", y = "Predicted Probability") +
  guides(colour = guide_legend(title = "Brexit Vote"),
         fill = guide_legend(title = "Brexit Vote")) +
  theme_bw() +
  scale_fill_viridis_d() +
  scale_colour_viridis_d() +
  facet_grid(response.level ~., scales = "free")
```



This plot is way more fun than the one above.

Let's discuss each plot in turn starting with the top plot (`Conservative`),

> *The predicted probability of identifying with the Conservative Party increases as age increases for both Remain and Leave voters. Leave voters are the most likely to identify with the Conservative Party across all ages.*

We might discuss the middle plot (`Other`) as the following,

> *The predicted probability of identifying with a party different from the Conservative Party or UKIP/Brexit Party decreases as age increases for both Remain and Leave voters. Remain voters*

are the most likely to identify with a different party across all ages.

Lastly, the bottom plot (`UKIP Brexit`),

> *The predicted probability of identifying with the UKIP/Brexit Party increases as age increases but only for Leave voters. The predicted probability is roughly flat across all ages for Remain voters. Also, notice the condensed values on the y-axis which means that the observed probability increase for Leave voters is not as large as the changes in the other two plots.*