Homework-4(Total Marks=49/50(All the results mathces the ones for Prof. Interpretations are reasonbly done))

Challenge: Explore with `party`

Dancun Juma

2025-02-19

Contents

Load the necessary libraries

Data Import

Fit the model that also includes `party` and discuss differences between the above model and this model with the additional predictor variable. Can you assess (think back to the MLR activity for how we tested two models where one was a subset of another) the effect by including this additional predictor variable?

The Data

Today we will analyze data from an online Ipsos (a consulting firm) survey that was conducted for a `FiveThirthyEight` article Why Many Americans Don't Vote. You can read more about the survey design and respondents in the `README` of their GitHub repo for the data.

Briefly, respondents were asked a variety of questions about their political beliefs, thoughts on multiple issues, and voting behavior. We will focus on the demographic variables and the respondent's party identification to understand whether a person is a probable voter (with levels always, sporadic, rarely/never).

The specific variables we will use are (definitions are from the `nonvoters_codebook.pdf`):

- `ppage`: Age of respondent
- `educ`: Highest educational attainment category
- `race`: Race of respondent, census categories Note: all categories except Hispanic are non-Hispanic
- `gender`: Gender of respondent
- `income_cat`: Household income category of respondent
- `Q30`: Response to the question "Generally speaking, do you think of yourself as a…"
  - 1: Republican
  - 2: Democrat
  - 3: Independent
  - 4: Another party, please specify
  - 5: No preference
  - -1: No response
- `voter_category`: past voting behavior:
  - **always**: respondent voted in all or all-but-one of the elections they were eligible in
  - **sporadic**: respondent voted in at least two, but fewer than all-but-one of the elections they were eligible in
  - **rarely/never**: respondent voted in 0 or 1 of the elections they were eligible in

These data can be read from the `nonvoters.csv` file this folder and were originally downloaded from: `https://github.com/fivethirtyeight/data/tree/master/non-voters`

Notes:

- Similarly to the data you used for the logistic regression portion of this activity, the researchers have the variable labeled `gender`, but it is unclear how this question was asked or what categorizations (if any) were provided to respondents to select from. We will use this as, "individuals that chose to provide their gender."
- The authors use weighting to make the final sample more representative on the US population for their article. We will **not** use weighting in this activity, so we will treat the sample as a convenience sample rather than a random sample of the population.

Now…

- Below, create a new R code chunk and write the code to:
  - Load `{tidyverse}` and `{tidymodels}` and any other packages you want to use.
  - *Read* in the *CSV* file from the `data` folder and store it in an R dataframe called `nonvoters`.
  - `select` only the variables listed above to want to make viewing/managing the data (and the `augment` output later) easier.

- Give your R code chunk a meaningful name, then run your code chunk.

```
# Read data and select relevant variables
nonvoters <- read_csv("nonvoters.csv") %>%
  select(ppage, educ, race, gender, income_cat, Q30, voter_category) %>%
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    income_cat = as.factor(income_cat),
    educ = as.factor(educ),
    voter_category = factor(voter_category, levels = c("rarely/never", "sporadic", "always")),
    party = case_when(
      Q30 == 1 ~ "Republican",
      Q30 == 2 ~ "Democrat",
      Q30 == 3 ~ "Independent",
      TRUE     ~ "Other"
    ) %>% as.factor()
  ) %>%
  filter(!is.na(party))
```

```
## Rows: 5836 Columns: 119
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): educ, race, gender, income_cat, voter_category
## dbl (114): RespId, weight, Q1, Q2_1, Q2_2, Q2_3, Q2_4, Q2_5, Q2_6, Q2_7, Q2_...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

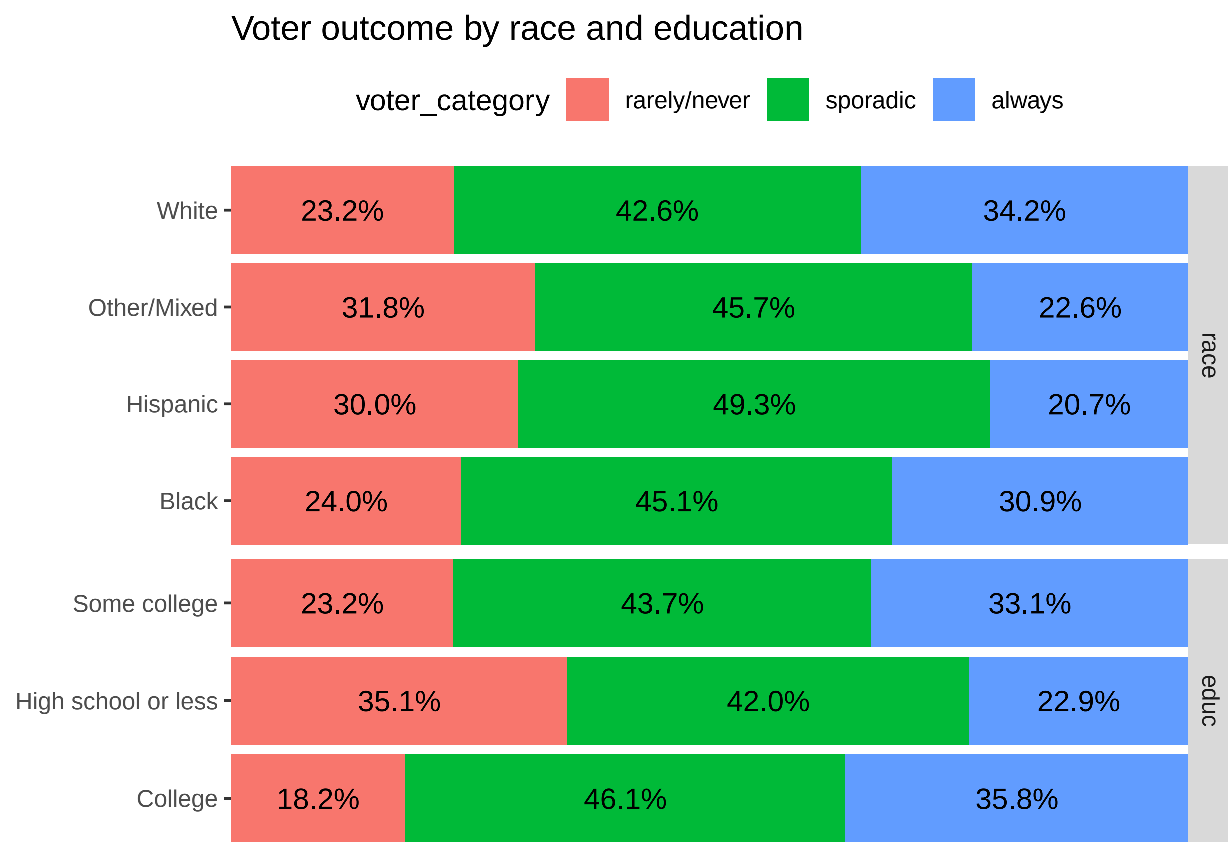
```
# Display the first few rows
head(nonvoters)
```

```
## # A tibble: 6 x 8
##   ppage educ                 race  gender income_cat     Q30 voter_category party
##   <dbl> <fct>                <fct> <fct>  <fct>        <dbl> <fct>          <fct>
## 1    73 College              White Female $75-125k         2 always         Demo~
## 2    90 College              White Female $125k or mo~     3 always         Inde~
## 3    53 College              White Male   $125k or mo~     2 sporadic       Demo~
## 4    58 Some college         Black Female $40-75k          2 sporadic       Demo~
## 5    81 High school or less  White Male   $40-75k          1 always         Repu~
## 6    61 High school or less  White Female $40-75k          5 rarely/never   Other
```

After doing this, answer the following questions: 1. Why do you think the authors chose to only include data from people who were eligible to vote for at least four election cycles? **The authors included only those eligible for at least four election cycles to ensure stable voting behavior classifications, reduce age-related bias, allow meaningful trend comparisons, and improve predictive power. This approach excludes short-term fluctuations and ensures respondents had multiple opportunities to vote, making the analysis more reliable and representative of long-term voting habits.**

2. In the FiveThirtyEight article, the authors include visualizations of the relationship between the voter category and demographic variables. Select two of these demographic variables. Then, for each variable, create and interpret a plot to describe its relationship with `voter_category`.

```
nonvoters %>%
  ggbivariate("voter_category",c("race","educ"),
            title="Voter outcome by race and education")
```



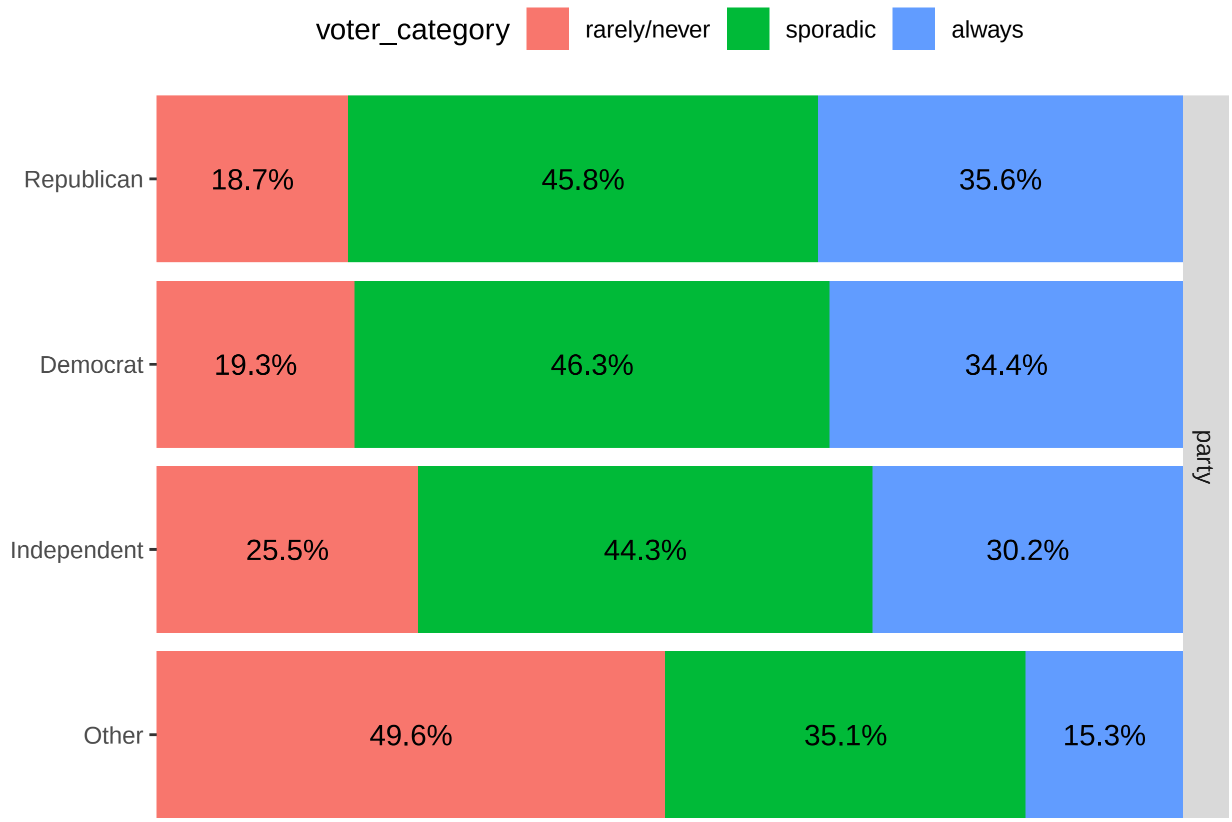
We need to do some data preparation before we fit our multinomial logistic regression model.

- Create a new R code chunk and address these items:
  - The variable `Q30` contains the respondent's political party identification. *Create a new variable* called `party` in the dataset that simplifies `Q30` into four categories: "Democrat", "Republican", "Independent", "Other" ("Other" should also include respondents who did not answer the question).
  - The variable `voter_category` identifies the respondent's past voter behavior. *Convert* this to a factor variable and ensure (*hint*: explore `relevel`) that the "rarely/never" level is the baseline level, followed by "sporadic", then "always".
- Then, run your code chunk

```
nonvoters %>%
  mutate(party = case_match(Q30,
                            1 ~ "Republican",
                            2 ~ "Democrat",
                            3 ~ "Independent",
                            c(4,5,-1) ~ "Other"),
         party = factor(party,
                        levels=c("Other","Independent","Democrat","Republican")),
         voter_category = factor(voter_category,
                                 levels=c("rarely/never","sporadic","always"))) -> nonvoters
```

Check that your changes are correct by creating a stacked bar graph using your new `Q30` variable as the y-axis and the `voter_category` represented with different colors. **Challenge**: Can you use the same color palette (*hint*: this is a handy tool, https://pickcoloronline.com/) that `FiveThirtyEight` used in their article?

```
nonvoters %>%
  ggbivariate("voter_category","party")
```



Fitting the model

Previously, we have explored logistic regression where the outcome/response/independent variable has two levels (e.g., "has feature" and "does not have feature"). We then used the logistic regression model

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

Another way to think about this model is if we are interested in comparing our "has feature" category to the *baseline* "does not have feature" category. If we let $y = 0$ represent the *baseline category*, such that $P(y_i = 1|X's) = \hat{p}_i 1$ and $P(y_i = 0|X's) = 1 - \hat{p}_{i1} = \hat{p}_{i0}$, then the above equation can be rewritten as:

$$\log\left(\frac{\hat{p}_{i1}}{\hat{p}_{i0}}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}$$

Recall that:

- The slopes $(\hat{\beta}_p)$ represent when x_p increases by one (x_p) unit, the odds of $y = 1$ compared to the baseline $y = 0$ are expected to multiply by a factor of $e^{\hat{\beta}_p}$. -The intercept $(\hat{\beta}0)$ respresents when all $x_j = 0$ (for $j = 1, \ldots, p$), the predicted odds of $y = 1$ versus the baseline $y = 0$ are $e^{\hat{\beta}_0}$.

For a multinomial (i.e., more than two categories, say, labeled $k = 1, 2, \ldots, K$) outcome variable, $P(y = 1) = p_1, P(y = 2) = p_2, \ldots, P(y = K) = p_k$, such that

$$\sum_{k=1}^{K} p_k = 1$$

This is called the **multinomial distribution**.

For a multinomial logistic regression model it is helpful to identify a baseline category (say, $y = 1$). We then fit a model such that $P(y = k) = p_k$ is a model of the x's.

$$\log\left(\frac{\hat{p}_{ik}}{\hat{p}_{i1}}\right) = \hat{\beta}_{0k} + \hat{\beta}_{1k}x_{i1} + \hat{\beta}_{2k}x_{i2} + \cdots + \hat{\beta}_{pk}x_{ip}$$

Notice that for a multinomial logistic model, we will have separate equations for each category of the outcome variable **relative to the baseline category**. If the outcome has K possible categories, there will be $K - 1$ equations as part of the multinomial logistic model.

Suppose we have an outcome variable y with three possible levels coded as "A", "B", "C". If "A" is the baseline category, then

$$\log\left(\frac{\hat{p}_{iB}}{\hat{p}_{iA}}\right) = \hat{\beta}_{0B} + \hat{\beta}_{1B}x_{i1} + \hat{\beta}_{2B}x_{i2} + \cdots + \hat{\beta}_{pB}x_{ip}$$
$$\log\left(\frac{\hat{p}_{iC}}{\hat{p}_{iA}}\right) = \hat{\beta}_{0C} + \hat{\beta}_{1C}x_{i1} + \hat{\beta}_{2C}x_{i2} + \cdots + \hat{\beta}_{pC}x_{ip}$$

Now we will fit a model using age, race, gender, income, and education to predict voter category. This is using `{tidymodels}`.

- In the code chunk below, replace "verbatim" with "r",
- Provide the code chunk a meaningful name/title, then run it.

```
# abbreviated recipe from previous activities
multi_mod <- multinom_reg() %>%
  set_engine("nnet") %>%
  fit(voter_category ~ ppage + educ + race + gender + income_cat, data = nonvoters)

tidy(multi_mod) %>%
  print(n = Inf) # This will display all rows of the tibble
```

```
## # A tibble: 22 x 6
##    y.level  term                     estimate std.error statistic   p.value
##    <chr>    <chr>                       <dbl>     <dbl>     <dbl>     <dbl>
##  1 sporadic (Intercept)              -0.887      0.167    -5.32    1.03e-  7
##  2 sporadic ppage                     0.0476     0.00229  20.8     5.05e- 96
##  3 sporadic educHigh school or less  -0.922      0.0957   -9.64    5.42e- 22
##  4 sporadic educSome college         -0.357      0.0938   -3.81    1.40e-  4
##  5 sporadic raceHispanic             -0.00655    0.126    -0.0521  9.58e-  1
##  6 sporadic raceOther/Mixed          -0.373      0.157    -2.38    1.74e-  2
##  7 sporadic raceWhite                -0.127      0.102    -1.25    2.10e-  1
##  8 sporadic genderMale               -0.0961     0.0707   -1.36    1.74e-  1
##  9 sporadic income_cat$40-75k        -0.127      0.110    -1.15    2.48e-  1
## 10 sporadic income_cat$75-125k       -0.000882   0.106    -0.00832 9.93e-  1
## 11 sporadic income_catLess than $40k -0.663      0.112    -5.91    3.43e-  9
## 12 always   (Intercept)              -1.85       0.185   -10.0     1.18e- 23
## 13 always   ppage                     0.0606     0.00252  24.0     2.29e-127
## 14 always   educHigh school or less  -1.35       0.107   -12.7     7.65e- 37
## 15 always   educSome college         -0.412      0.100    -4.10    4.13e-  5
## 16 always   raceHispanic             -0.417      0.147    -2.84    4.46e-  3
## 17 always   raceOther/Mixed          -0.683      0.182    -3.74    1.82e-  4
## 18 always   raceWhite                 0.0392     0.111     0.353   7.24e-  1
## 19 always   genderMale               -0.211      0.0779   -2.70    6.83e-  3
```

```
## 20 always   income_cat$40-75k        -0.0669     0.120    -0.559   5.76e-  1
## 21 always   income_cat$75-125k        0.147      0.113     1.29    1.95e-  1
## 22 always   income_catLess than $40k -0.756      0.125    -6.05    1.43e-  9
```

`{tidymodels}` is designed for cross-validation and so there needs to be some "trickery" when we build models using the entire dataset. For example, when you type `multi_mod$fit$call` in your **Console**, you should see the following output:

```
multi_mod$fit$call
```

```
## nnet::multinom(formula = voter_category ~ ppage + educ + race +
##     gender + income_cat, data = data, trace = FALSE)
```

The issue here is `data = data` and should be `data = nonvoters`. To *repair* this, add the following to your previous R code chunk:

```
multi_mod <- repair_call(multi_mod, data = nonvoters)
```

Re-run your code chunk, then type `multi_mod$fit$call` in your **Console**, you should see the following output:

```
multi_mod$fit$call
```

```
## nnet::multinom(formula = voter_category ~ ppage + educ + race +
##     gender + income_cat, data = nonvoters, trace = FALSE)
```

```
nnet::multinom(formula = voter_category ~ ppage + educ + race + gender + income_cat, data = nonvoters, 
```

```
## Call:
## nnet::multinom(formula = voter_category ~ ppage + educ + race +
##     gender + income_cat, data = nonvoters, trace = FALSE)
##
## Coefficients:
##          (Intercept)      ppage educHigh school or less educSome college
## sporadic  -0.8871838 0.04756772              -0.9221942       -0.3570387
## always    -1.8538235 0.06059776              -1.3530192       -0.4119994
##          raceHispanic raceOther/Mixed   raceWhite  genderMale income_cat$40-75k
## sporadic -0.006553032      -0.3728841 -0.12740919 -0.09612107       -0.12721356
## always   -0.417200524      -0.6827372  0.03923864 -0.21058506       -0.06688936
##          income_cat$75-125k income_catLess than $40k
## sporadic      -0.0008817095               -0.6625834
## always         0.1466082776               -0.7563449
##
## Residual Deviance: 11386.63
## AIC: 11430.63
```

Yay!

Now, recall that the baseline category for the model is `"rarely/never"`. Using your `tidy(multi_mod) %>% print(n = Inf)` output, complete the following items:

3. Write the model equation for the log-odds of a person that the "rarely/never" votes vs "always" votes.

```
tidy(multi_mod) %>%
  filter(y.level == 'always') %>%
  select(estimate, term)
```

```
## # A tibble: 11 x 2
##    estimate term
##       <dbl> <chr>
```

```
## 1  -1.85   (Intercept)
## 2   0.0606 ppage
## 3  -1.35   educHigh school or less
## 4  -0.412  educSome college
## 5  -0.417  raceHispanic
## 6  -0.683  raceOther/Mixed
## 7   0.0392 raceWhite
## 8  -0.211  genderMale
## 9  -0.0669 income_cat$40-75k
## 10  0.147  income_cat$75-125k
## 11 -0.756  income_catLess than $40k
```

That is, finish this equation using your estimated parameters:

$$
\begin{aligned}
\log\left(\frac{\hat{p}_{\text{"always"}}}{\hat{p}_{\text{"rarely/never"}}}\right) = -1.8538 & \\
&+ 0.0606 \cdot \text{ppage} \\
&- 1.3530 \cdot \text{educHigh school or less} \\
&- 0.4120 \cdot \text{educSome college} \\
&- 0.4172 \cdot \text{raceHispanic} \\
&- 0.6827 \cdot \text{raceOther/Mixed} \\
&+ 0.0392 \cdot \text{raceWhite} \\
&- 0.2106 \cdot \text{genderMale} \\
&- 0.0669 \cdot \text{income\_cat \$40-75k} \\
&+ 0.1466 \cdot \text{income\_cat \$75-125k} \\
&- 0.7563 \cdot \text{income\_cat Less than \$40k}
\end{aligned}
$$

4. For your equation in (3), interpret the slope for `genderMale` in both log-odds and odds.

The coefficient -0.2106 means that, holding all other variables constant, the log-odds of a male being in the "always" voter category instead of the "rarely/never" category decreases by 0.2106 compared to a female. In other words, being male is associated with a lower likelihood of always voting relative to rarely/never voting.

Note: The interpretation for the slope for `ppage` is a little more difficult to interpret. However, we could mean-center age (i.e., subtract the mean age from each age value) to have a more meaningful interpretation.

Predicting

We could use this model to calculate probabilities. Generally, for categories $2, \ldots, K$, the probability that the i^{th} observation is in the k^{th} category is,

$$\hat{p}_{ik} = \frac{e^{\hat{\beta}_{0j} + \hat{\beta}_{1j}x_{i1} + \hat{\beta}_{2j}x_{i2} + \cdots + \hat{\beta}_{pj}x_{ip}}}{1 + \sum_{k=2}^{K} e^{\hat{\beta}_{0k} + \hat{\beta}_{1k}x_{1i} + \hat{\beta}_{2k}x_{2i} + \cdots + \hat{\beta}_{pk}x_{pi}}}$$

And the baseline category, $k = 1$,

$$\hat{p}_{i1} = 1 - \sum_{k=2}^{K} \hat{p}_{ik}$$

However, we will let R do these calculations.

- In the code chunk below, replace "verbatim" with "r",
- Provide the code chunk a meaningful name/title, then run it.

```
voter_aug <- augment(multi_mod, new_data = nonvoters)

voter_aug
```

```
## # A tibble: 5,836 x 12
##    .pred_class `.pred_rarely/never` .pred_sporadic .pred_always ppage educ
##    <fct>                      <dbl>          <dbl>        <dbl> <dbl> <fct>
##  1 always                    0.0352          0.411        0.554    73 College
##  2 always                    0.0153          0.402        0.583    90 College
##  3 sporadic                  0.119           0.489        0.391    53 College
##  4 sporadic                  0.121           0.485        0.394    58 Some coll~
##  5 sporadic                  0.0930          0.505        0.402    81 High scho~
##  6 sporadic                  0.204           0.472        0.324    61 High scho~
##  7 sporadic                  0.0778          0.504        0.418    80 High scho~
##  8 sporadic                  0.102           0.515        0.382    68 Some coll~
##  9 sporadic                  0.0516          0.475        0.474    70 College
## 10 always                    0.0380          0.454        0.508    83 Some coll~
## # i 5,826 more rows
## # i 6 more variables: race <fct>, gender <fct>, income_cat <fct>, Q30 <dbl>,
## #   voter_category <fct>, party <fct>
```

```
voter_aug %>%
  select(contains("pred"))
```

```
## # A tibble: 5,836 x 4
##    .pred_class `.pred_rarely/never` .pred_sporadic .pred_always
##    <fct>                      <dbl>          <dbl>        <dbl>
##  1 always                    0.0352          0.411        0.554
##  2 always                    0.0153          0.402        0.583
##  3 sporadic                  0.119           0.489        0.391
##  4 sporadic                  0.121           0.485        0.394
##  5 sporadic                  0.0930          0.505        0.402
##  6 sporadic                  0.204           0.472        0.324
##  7 sporadic                  0.0778          0.504        0.418
##  8 sporadic                  0.102           0.515        0.382
##  9 sporadic                  0.0516          0.475        0.474
## 10 always                    0.0380          0.454        0.508
## # i 5,826 more rows
```

Here we can see all of the predicted probabilities. This is still rather difficult to view so a confusion matrix can help us summarize how well the predictions fit the actual values.

- In the code chunk below, replace "verbatim" with "r",
- Provide the code chunk a meaningful name/title, then run it.

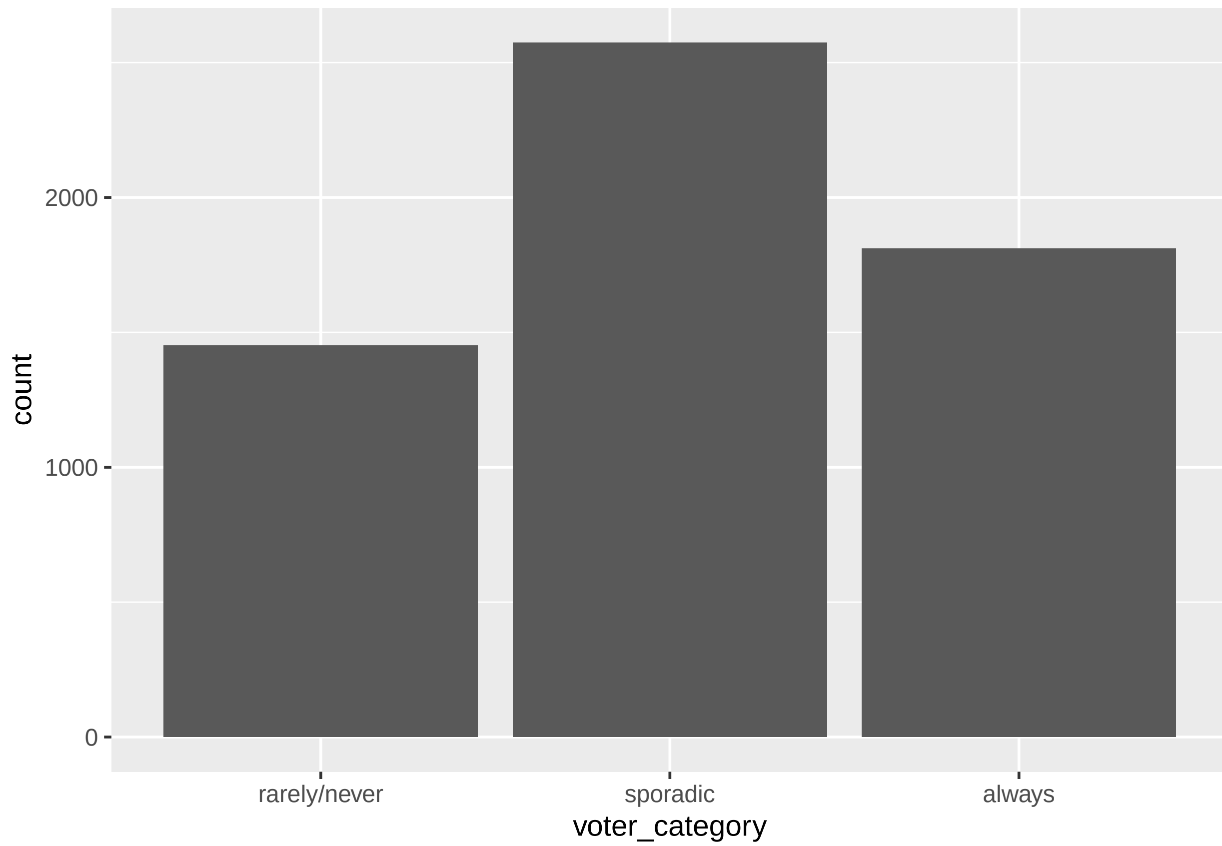
```
voter_conf_mat <- voter_aug %>%
  count(voter_category, .pred_class, .drop = FALSE)

voter_conf_mat %>%
  pivot_wider(
    names_from = .pred_class,
    values_from = n
  )
```

```
## # A tibble: 3 x 4
##   voter_category `rarely/never` sporadic always
##   <fct>                   <int>    <int>  <int>
## 1 rarely/never              586      815     50
## 2 sporadic                  271     1994    309
## 3 always                    243     1150    418
```

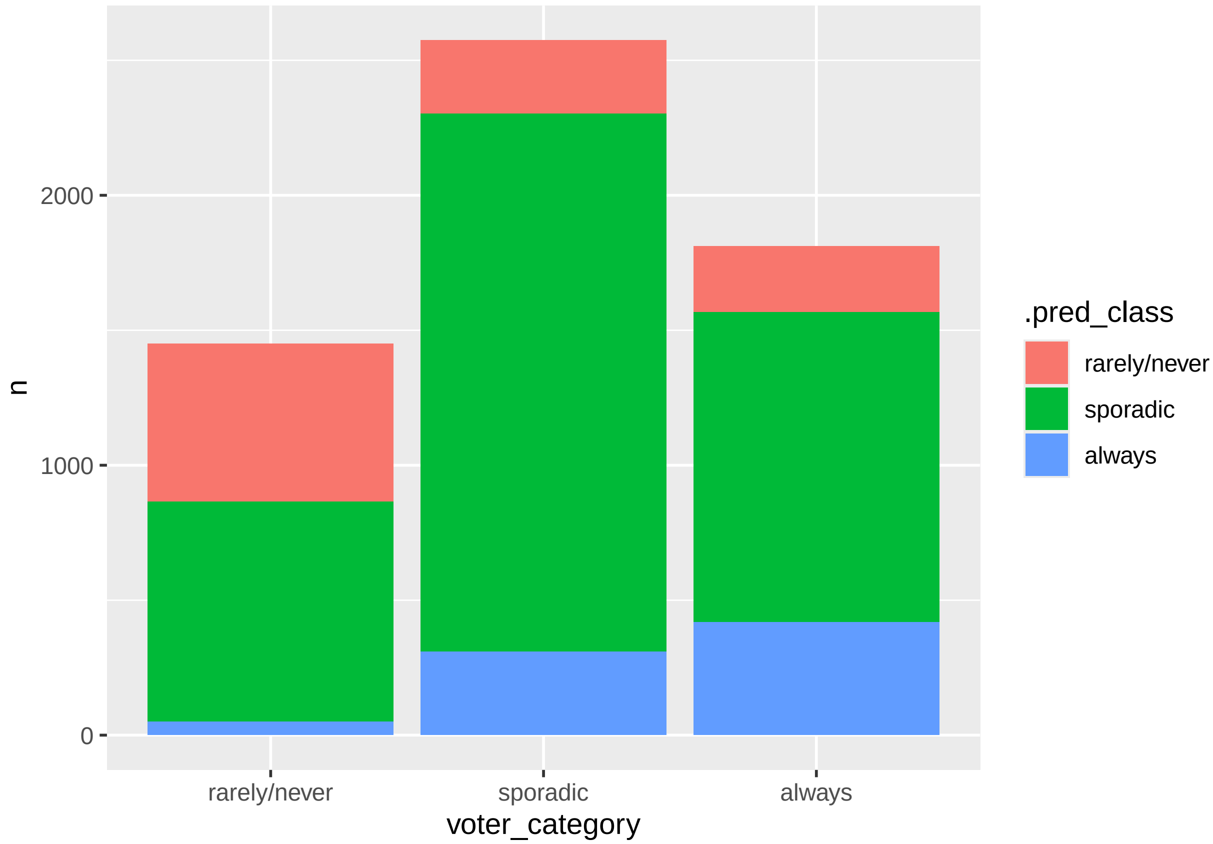
We can also visualize how well these predictions fit the original values.

- In the code chunk below, replace "verbatim" with "r",
- Provide the code chunk a meaningful name/title, then run it.

```
nonvoters %>%
  ggplot(aes(x = voter_category)) +
  geom_bar() +
  labs(
    main = "Self-reported voter category"
    )
```




```
voter_conf_mat %>%
  ggplot(aes(x = voter_category, y = n, fill = .pred_class)) +
  geom_bar(stat = "identity") +
  labs(
    main = "Predicted vs self-reported voter category"
    )
```



Answer the following question:

5. What do you notice?

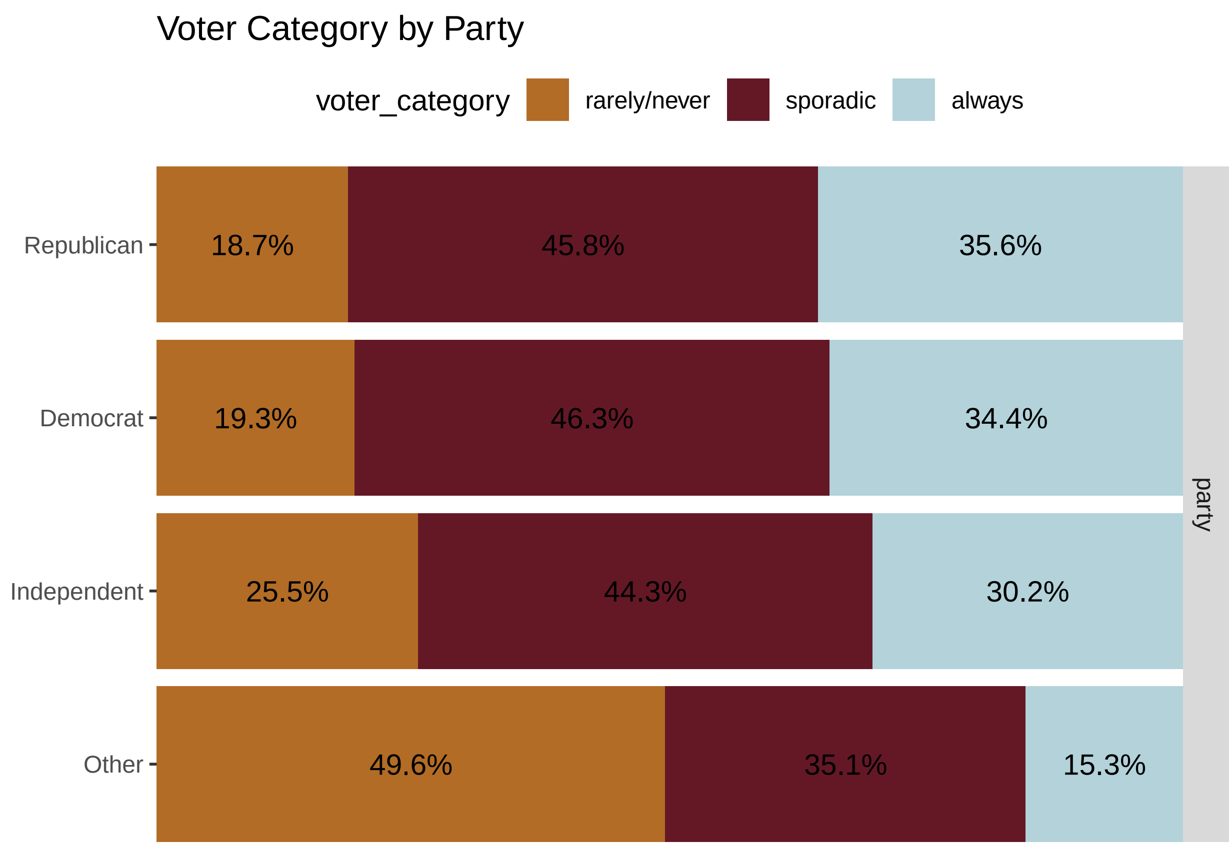
Better prediction

Challenge: Explore with `party`

Fit the model that also includes `party` and discuss differences between the above model and this model with the additional predictor variable. Can you assess (think back to the MLR activity for how we tested two models where one was a subset of another) the effect by including this additional predictor variable?

Part 1: Create a bivariate plot of voter_category by party. What does it suggest about the predictive capability of party for voting?

```
# Define FiveThirtyEight color palette
fte_colors <- c("rarely/never" = "#b26c26",
                "sporadic" = "#641826",
                "always" = "#b3d2da")

# Bivariate plot: voter_category by party
nonvoters %>%
  ggbivariate("voter_category",c("party"),
              title="Voter Category by Party")+
  scale_fill_manual(values = fte_colors)
```



Colors were obtained from this link "https://pickcoloronline.com/#google_vignette".

The plot suggests that party affiliation has `some` predictive capability for voting behavior but is not a strong determinant. While differences exist such as a higher proportion of `always` voters among major parties (Democrat, Republican) and more `rarely/never` voters in the `Other` category—each party contains a mix of voter types. This indicates that while party affiliation may influence voting frequency, other factors (e.g., demographics, engagement, or political interest) likely play a significant role. In general, party alone does not strongly predict voter turnout, suggesting a need for more nuanced variables to improve predictive accuracy.

Part 2: Write out the new equation for always with respect to rarely/never.

Equation format of the model

```
# Fit multinomial regression model with party
multi_mod_party <- multinom_reg() %>%
  set_engine("nnet") %>%
  fit(voter_category ~ ppage + educ + race + gender + income_cat + party, data = nonvoters, trace = FALS

tidy(multi_mod_party) %>%
  print(n = Inf)
```

```
## # A tibble: 28 x 6
##    y.level  term                          estimate std.error statistic   p.value
##    <chr>    <chr>                            <dbl>     <dbl>     <dbl>     <dbl>
```

```
## 1 sporadic (Intercept)              -1.57      0.188    -8.34    7.58e- 17
## 2 sporadic ppage                     0.0457    0.00232  19.7     5.33e- 86
## 3 sporadic educHigh school or less  -0.853     0.0974   -8.76    1.94e- 18
## 4 sporadic educSome college         -0.293     0.0952   -3.08    2.10e-  3
## 5 sporadic raceHispanic              0.0402    0.128     0.314   7.53e-  1
## 6 sporadic raceOther/Mixed          -0.332     0.159    -2.09    3.66e-  2
## 7 sporadic raceWhite                -0.0775    0.108    -0.719   4.72e-  1
## 8 sporadic genderMale               -0.0901    0.0722   -1.25    2.12e-  1
## 9 sporadic income_cat$40-75k        -0.0738    0.111    -0.662   5.08e-  1
## 10 sporadic income_cat$75-125k       0.0125    0.107     0.117   9.07e-  1
## 11 sporadic income_catLess than $40k -0.588    0.114    -5.17    2.36e-  7
## 12 sporadic partyIndependent         0.548     0.111     4.95    7.27e-  7
## 13 sporadic partyDemocrat            0.940     0.106     8.85    8.51e- 19
## 14 sporadic partyRepublican          0.857     0.112     7.62    2.61e- 14
## 15 always   (Intercept)             -2.92      0.218   -13.4     6.73e- 41
## 16 always   ppage                    0.0582    0.00257  22.7     5.94e-114
## 17 always   educHigh school or less -1.27      0.109   -11.6     2.46e- 31
## 18 always   educSome college        -0.330     0.102    -3.23    1.26e-  3
## 19 always   raceHispanic            -0.341     0.150    -2.28    2.28e-  2
## 20 always   raceOther/Mixed         -0.600     0.185    -3.24    1.20e-  3
## 21 always   raceWhite                0.127     0.119     1.07    2.84e-  1
## 22 always   genderMale              -0.192     0.0797   -2.41    1.58e-  2
## 23 always   income_cat$40-75k       -0.000380  0.121    -0.00313 9.97e-  1
## 24 always   income_cat$75-125k       0.165     0.114     1.45    1.48e-  1
## 25 always   income_catLess than $40k -0.664    0.127    -5.23    1.71e-  7
## 26 always   partyIndependent         0.839     0.136     6.15    7.52e- 10
## 27 always   partyDemocrat            1.40      0.131    10.7     1.29e- 26
## 28 always   partyRepublican          1.24      0.136     9.10    8.87e- 20
```

Most predictors were significant at p = 0.05

```r
multi_mod_party <- repair_call(multi_mod_party, data = nonvoters)
```

```r
multi_mod_party$fit$call
```

```
## nnet::multinom(formula = voter_category ~ ppage + educ + race +
##     gender + income_cat + party, data = nonvoters, trace = FALSE)
```

```r
nnet::multinom(formula = voter_category ~ ppage + educ + race + gender + income_cat + party, data = non
```

```
## Call:
## nnet::multinom(formula = voter_category ~ ppage + educ + race +
##     gender + income_cat + party, data = nonvoters, trace = FALSE)
##
## Coefficients:
##          (Intercept)      ppage educHigh school or less educSome college
## sporadic   -1.570548 0.04568563              -0.8532746       -0.2928771
## always     -2.919557 0.05820720              -1.2671996       -0.3303179
##          raceHispanic raceOther/Mixed   raceWhite genderMale income_cat$40-75k
## sporadic   0.04022428      -0.3324437 -0.07754206 -0.0900604     -0.0737627930
## always    -0.34100621      -0.6004728  0.12719733 -0.1922012     -0.0003797663
##          income_cat$75-125k income_catLess than $40k partyIndependent
## sporadic         0.01249208               -0.5878188        0.5480214
## always           0.16519070               -0.6641096        0.8387114
##          partyDemocrat partyRepublican
## sporadic     0.9404516       0.8566422
```

```
## always       1.4010026       1.2392412
##
## Residual Deviance: 11232.78
## AIC: 11288.78
```

Higher education and income increase the likelihood of always voting, while being male slightly reduces it. Party affiliation strongly impacts voting behavior, with Democrats and Republicans more likely to vote consistently. The model's AIC is 11288.78, indicating fit quality.

```
tidy(multi_mod_party) %>%
  filter(y.level=='always') %>%
  select(estimate, term)
```

```
## # A tibble: 14 x 2
##     estimate term
##        <dbl> <chr>
##  1 -2.92     (Intercept)
##  2  0.0582   ppage
##  3 -1.27     educHigh school or less
##  4 -0.330    educSome college
##  5 -0.341    raceHispanic
##  6 -0.600    raceOther/Mixed
##  7  0.127    raceWhite
##  8 -0.192    genderMale
##  9 -0.000380 income_cat$40-75k
## 10  0.165    income_cat$75-125k
## 11 -0.664    income_catLess than $40k
## 12  0.839    partyIndependent
## 13  1.40     partyDemocrat
## 14  1.24     partyRepublican
```

(Intercept) (-2.92): The baseline log-odds of always voting when all predictors are at reference levels.

Age (0.058): Older individuals are more likely to always vote.

Education - High school or less (-1.267): Those with only a high school education are much less likely to always vote.

Education - Some college (-0.330): Individuals with some college education are also less likely to always vote, but the effect is smaller.

Race - Hispanic (-0.341): Hispanic individuals are less likely to always vote.

Race - Other/Mixed (-0.600): People of mixed or other racial backgrounds are even less likely to always vote.

Race - White (0.127): White individuals are slightly more likely to always vote.

Gender - Male (-0.192): Males are less likely to always vote compared to females.

Income - $40-75k (-0.0004): This income group has almost no effect on voting behavior.

Income - $75-125k (0.165): Higher income increases the likelihood of always voting.

Income - Less than $40k (-0.664): Lower-income individuals are significantly less likely to always vote.

Party - Independent (0.839): Independents are more likely to always vote than those with no party preference.

Party - Democrat (1.401): Democrats are much more likely to always vote.

Party - Republican (1.239): Republicans are also more likely to always vote, though slightly less than Democrats.

Final equation

$$
\begin{aligned}
\log\left(\frac{\hat{p}_{\text{"always"}}}{\hat{p}_{\text{"rarely/never"}}}\right) = &-2.9196 \\
&+ 0.0582 \cdot \text{ppage} \\
&- 1.2672 \cdot \text{educHigh school or less} \\
&- 0.3303 \cdot \text{educSome college} \\
&- 0.3410 \cdot \text{raceHispanic} \\
&- 0.6005 \cdot \text{raceOther/Mixed} \\
&+ 0.1272 \cdot \text{raceWhite} \\
&- 0.1922 \cdot \text{genderMale} \\
&- 0.0004 \cdot \text{income\_cat \$40-75k} \\
&+ 0.1652 \cdot \text{income\_cat \$75-125k} \\
&- 0.6641 \cdot \text{income\_cat Less than \$40k} \\
&+ 0.8387 \cdot \text{partyIndependent} \\
&+ 1.4010 \cdot \text{partyDemocrat} \\
&+ 1.2392 \cdot \text{partyRepublican}
\end{aligned}
$$

Before controlling for political party, men were significantly less likely to be "always" voters than women (coefficient = -0.211). After controlling for party, the effect became slightly weaker (-0.192), meaning that some of the gender effect was actually due to party differences. The odds ratio for men voting always slightly increased, meaning that some of the gap in voting habits between genders can be attributed to differences in party affiliation rather than gender alone.

Part 3: Interpret the slope for genderMale. How did it change (if any)?

```
tidy(multi_mod) %>%
  filter(term == "genderMale")
```

```
## # A tibble: 2 x 6
##   y.level  term       estimate std.error statistic p.value
##   <chr>    <chr>         <dbl>     <dbl>     <dbl>   <dbl>
## 1 sporadic genderMale  -0.0961    0.0707     -1.36 0.174
## 2 always   genderMale  -0.211     0.0779     -2.70 0.00683
```

```
tidy(multi_mod_party) %>%
  filter(term == "genderMale")
```

```
## # A tibble: 2 x 6
##   y.level  term       estimate std.error statistic p.value
##   <chr>    <chr>         <dbl>     <dbl>     <dbl>   <dbl>
## 1 sporadic genderMale  -0.0901    0.0722     -1.25  0.212
## 2 always   genderMale  -0.192     0.0797     -2.41  0.0158
```

The table above shows that the effect of being male on the outcome became slightly less negative when party affiliation was accounted for. The statistical significance of the estimate for always remained, though its p-value increased from **0.0068** to **0.0158**, suggesting a reduced but still significant effect. I can say, adding **party** slightly reduced the impact of gender but did not eliminate its significance.

Part 4: Interpret the slopes for the two major parties (Republican, Democratic). What does this tell us?

The slopes for the two major parties, Republican and Democratic, in the regression results indicate the relationship between party affiliation and the outcome variable, adjusting for other covariates. For the "sporadic" model, the slope for the Republican party is -0.0838, suggesting that identifying as a Republican is associated with a slight decrease in the outcome variable compared to the reference group. This relationship, however, is not statistically significant (p = 0.415), meaning that the effect is not reliably different from zero in this context. In the "always" model, the slope for the Republican party is -0.162, which indicates a stronger negative association with the outcome variable. Although this effect is also relatively small, the p-value of 0.142 suggests that it is not statistically significant either, meaning there is no clear evidence of a strong relationship between Republican affiliation and the outcome when compared to other groups. In summary, the data does not provide strong evidence that party affiliation significantly impacts the outcome for either party.

Part 5: Get predictions from your model and discuss what they mean.

```r
# Get predictions from the model
voter_aug <- augment(multi_mod_party, new_data = nonvoters)
voter_aug
```

```
## # A tibble: 5,836 x 12
##    .pred_class `.pred_rarely/never` .pred_sporadic .pred_always ppage educ
##    <fct>                      <dbl>          <dbl>        <dbl> <dbl> <fct>
##  1 always                    0.0281          0.394        0.578    73 College
##  2 always                    0.0208          0.423        0.556    90 College
##  3 sporadic                  0.0947          0.480        0.425    53 College
##  4 sporadic                  0.0924          0.482        0.425    58 Some coll~
##  5 sporadic                  0.0762          0.506        0.418    81 High scho~
##  6 sporadic                  0.353           0.436        0.212    61 High scho~
##  7 sporadic                  0.0679          0.506        0.426    80 High scho~
##  8 sporadic                  0.0782          0.504        0.417    68 Some coll~
##  9 always                    0.0467          0.474        0.480    70 College
## 10 always                    0.0463          0.466        0.488    83 Some coll~
## # i 5,826 more rows
## # i 6 more variables: race <fct>, gender <fct>, income_cat <fct>, Q30 <dbl>,
## #   voter_category <fct>, party <fct>
```

```r
# View predictions
voter_aug %>%
  select(contains("pred"))
```

```
## # A tibble: 5,836 x 4
##    .pred_class `.pred_rarely/never` .pred_sporadic .pred_always
##    <fct>                      <dbl>          <dbl>        <dbl>
##  1 always                    0.0281          0.394        0.578
##  2 always                    0.0208          0.423        0.556
##  3 sporadic                  0.0947          0.480        0.425
##  4 sporadic                  0.0924          0.482        0.425
##  5 sporadic                  0.0762          0.506        0.418
##  6 sporadic                  0.353           0.436        0.212
```

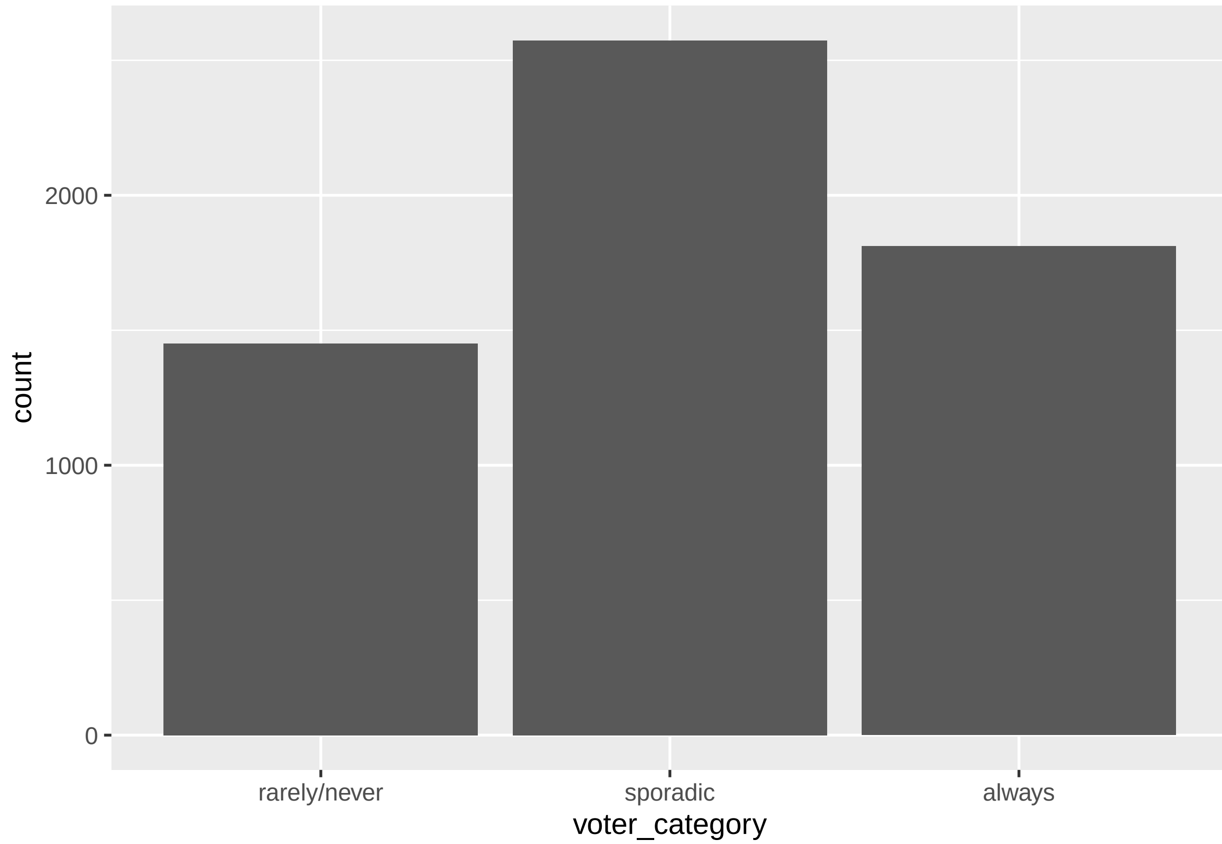
```
##  7 sporadic                  0.0679          0.506         0.426
##  8 sporadic                  0.0782          0.504         0.417
##  9 always                    0.0467          0.474         0.480
## 10 always                    0.0463          0.466         0.488
## # i 5,826 more rows
```

Taking for example the first row, the model predicts a 57.8% chance that the observation belongs to the "always" category, a 39.4% chance for "sporadic," and only a 2.8% chance for "rarely/never." This suggests that, for this particular case, the behavior is most likely to fall under the "always" category, with "sporadic" being the second most likely, and "rarely/never" a very small possibility.

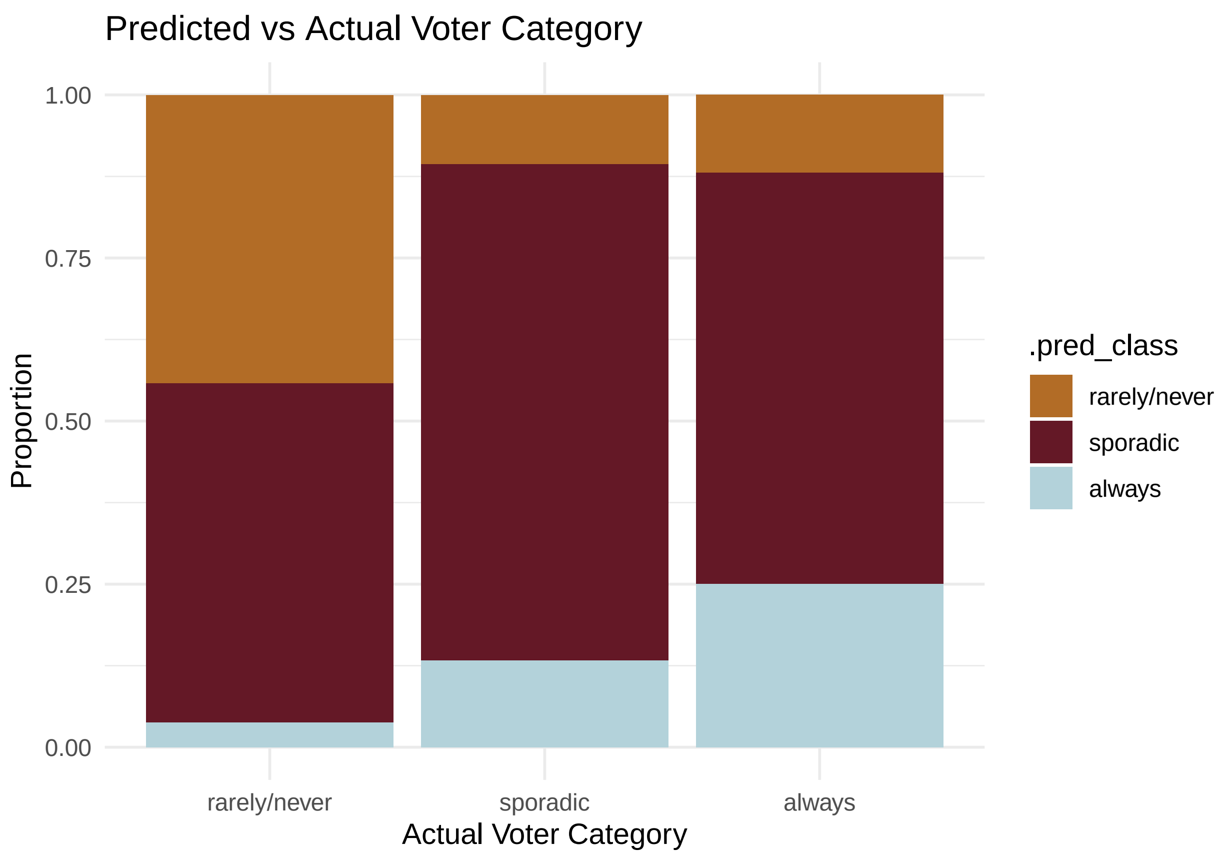
These predictions help me understand how likely certain behaviors are to occur based on the model's learning. They provide insights into the patterns of the data, which can be used to better tailor strategies or decisions related to these behaviors. By giving an example, let's say if we want to target individuals who show a consistent (always) pattern, we can focus on those with high probabilities in that category. On the other hand, analyzing those who are likely to fall into the "sporadic" group can help us adjust our approach for less consistent behaviors.

Part 6: Visualize the predictions versus the actual voter category and interpret it.

```r
nonvoters %>%
  ggplot(aes(x = voter_category)) +
  geom_bar() +
  labs(
    main = "Self-reported voter category"
    )
```




```
# Visualizing predicted vs actual voter category
voter_aug %>%
  ggplot(aes(x = voter_category, fill = .pred_class)) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = fte_colors)+
  labs(title = "Predicted vs Actual Voter Category",
       x = "Actual Voter Category",
       y = "Proportion",
       fill = ".pred_class") +
  theme_minimal()
```



From the plot here, I can say the model correctly predicts a high proportion of sporadic voters across all categories but struggles to distinguish between rarely/never and always voters. Misclassification occurs mostly in the rarely/never group, where many are predicted as sporadic, indicating potential overlap in characteristics.

Part 7: How did the confusion matrix change? Interpret each entry in the confusion matrix.

```
# Confusion matrix
voter_conf_mat <- voter_aug %>%
  count(voter_category, .pred_class, .drop = FALSE)

voter_conf_mat %>%
  pivot_wider(
    names_from = .pred_class,
    values_from = n
  )
```

```
## # A tibble: 3 x 4
##   voter_category `rarely/never` sporadic always
##   <fct>                   <int>    <int>  <int>
## 1 rarely/never              642      754     55
## 2 sporadic                  274     1958    342
## 3 always                    216     1142    453
```

Rarely/never (actual) vs. Rarely/never (predicted): 642 individuals who are “rarely/never” were correctly predicted as "rarely/never."

Rarely/never (actual) vs. Sporadic (predicted): 754 individuals who are "rarely/never" were incorrectly predicted as "sporadic."

Rarely/never (actual) vs. Always (predicted): 55 individuals who are "rarely/never" were incorrectly predicted as "always."

Sporadic (actual) vs. Rarely/never (predicted): 274 individuals who are "sporadic" were incorrectly predicted as "rarely/never."

Sporadic (actual) vs. Sporadic (predicted): 1958 individuals who are "sporadic" were correctly predicted as "sporadic."

Sporadic (actual) vs. Always (predicted): 342 individuals who are "sporadic" were incorrectly predicted as "always."

Always (actual) vs. Rarely/never (predicted): 216 individuals who always behave consistently were incorrectly predicted as "rarely/never."

Always (actual) vs. Sporadic (predicted): 1142 individuals who always behave consistently were incorrectly predicted as "sporadic."

Always (actual) vs. Always (predicted): 453 individuals who always behave consistently were correctly predicted as "always."

Change in Confusion Matrix for with party and without

Based on Correct predictions - The "sporadic" category continues to have the highest correct predictions (1994 and 1958), while "always" predictions are still somewhat low, but improved slightly in the with-party matrix (418 to 453).

Based on Misclassifications - The "rarely/never" category now has more individuals correctly predicted (642) compared to the previous 586, suggesting that party affiliation helps slightly reduce the misclassification of this group as "sporadic."

Based on Sporadic vs. Always - I see that party affiliation does not drastically change the misclassification between "sporadic" and "always," with some individuals still misclassified in both directions. For example, "always" voters are misclassified as "sporadic" (1150 vs. 1142) or "rarely/never" (243 vs. 216).

Part 8: Discuss the assessment of adding party suggested in the challenge. Did adding party help the model? Why or why not?

```r
# Assessing the effect of adding 'party'
# Compare models with and without 'party'
mod_no_party <- multinom(voter_category ~ ppage + educ + race + gender + income_cat, data = nonvoters, 
mod_with_party <- multinom(voter_category ~ ppage + educ + race + gender + income_cat + party, data = no

# Compare AIC values
AIC(mod_no_party, mod_with_party)
```

```
##                df      AIC
## mod_no_party   22 11430.63
## mod_with_party 28 11288.78
```

Assessment of Adding Party Affiliation to the Model

The results suggest that adding party affiliation to the model does have some impact, but it may not significantly improve the model's predictive power based on the outcomes we see in both the model with and without party affiliation.

Model Performance Comparison (BONUS explanation)

Without Party (mod_no_party): The model has a log-likelihood value of 22 and a residual deviance of 11430.63.

With Party (mod_with_party): The model has a log-likelihood value of 28 and a residual deviance of 11288.78.

The reduction in deviance from 11430.63 to 11288.78 suggests a small improvement in fit when party affiliation is included. The lower deviance indicates that the model with party affiliation explains the data slightly better.

Impact of Party on Coefficients

Looking at the coefficients for the "with party" model, I saw that some variables, like genderMale and raceHispanic, show more statistically significant relationships with the outcome compared to the model without party affiliation. For instance, the p-value for genderMale in the "with party" model is 0.00683, which indicates a statistically significant effect on behavior.

I can say the addition of party did not drastically change the relationships for educational level, but it did provide clearer insights into the impact of party on "sporadic" vs. "always" behavior (for example coefficients for party terms could improve predictability, but they aren't explicitly listed in the output provided).

Why Adding Party Helps (or Doesn't)

1. Small Improvement in Fit - The deviance drop suggests a slight improvement, but not a major one. The change in deviance isn't dramatic, indicating that while party affiliation contributes to the model, it may not drastically change the underlying behavior patterns.
2. Significance of Party Terms - Party affiliation can have a significant effect on behavior prediction, but this may not always translate to drastic model improvement in terms of overall fit. It can refine the predictions for certain subgroups, but the model's overall structure is still largely influenced by other variables like education, income, and race.
3. Misclassification Patterns - Despite the addition of party, the misclassification patterns (between "sporadic" and "always" behavior) remain somewhat similar, suggesting that other factors not included in the model might also be significant for explaining behavior consistency.

Conclusion

While adding party affiliation does provide a small improvement in model fit (as evidenced by the drop in deviance), it does not radically improve the model's predictive power or dramatically shift the coefficients. It suggests that party affiliation may be a relevant predictor, but it might not be the most influential one compared to other factors like education or income. The model might still benefit from further refinement or the inclusion of additional factors that capture the complexity of voter behavior more comprehensively.