Interactions & Model Building in Logistic Regression

Overview for Today

Today we will be learning about:

- Interpreting & testing interactions using logistic regression
- Testing competing models

Review of Logistic Regression

A few things to keep in mind:

- Use logistic regression when your outcome is binary.
- Transform your regression coefficients using the exponential function before interpreting.
- Transformed regression coefficients are **odds ratios**.
- Statistical testing is identical to linear regression

Who Votes?

You’re a policy analyst trying to predict voter turnout for a given region. You have the following information on voters for the previous election:

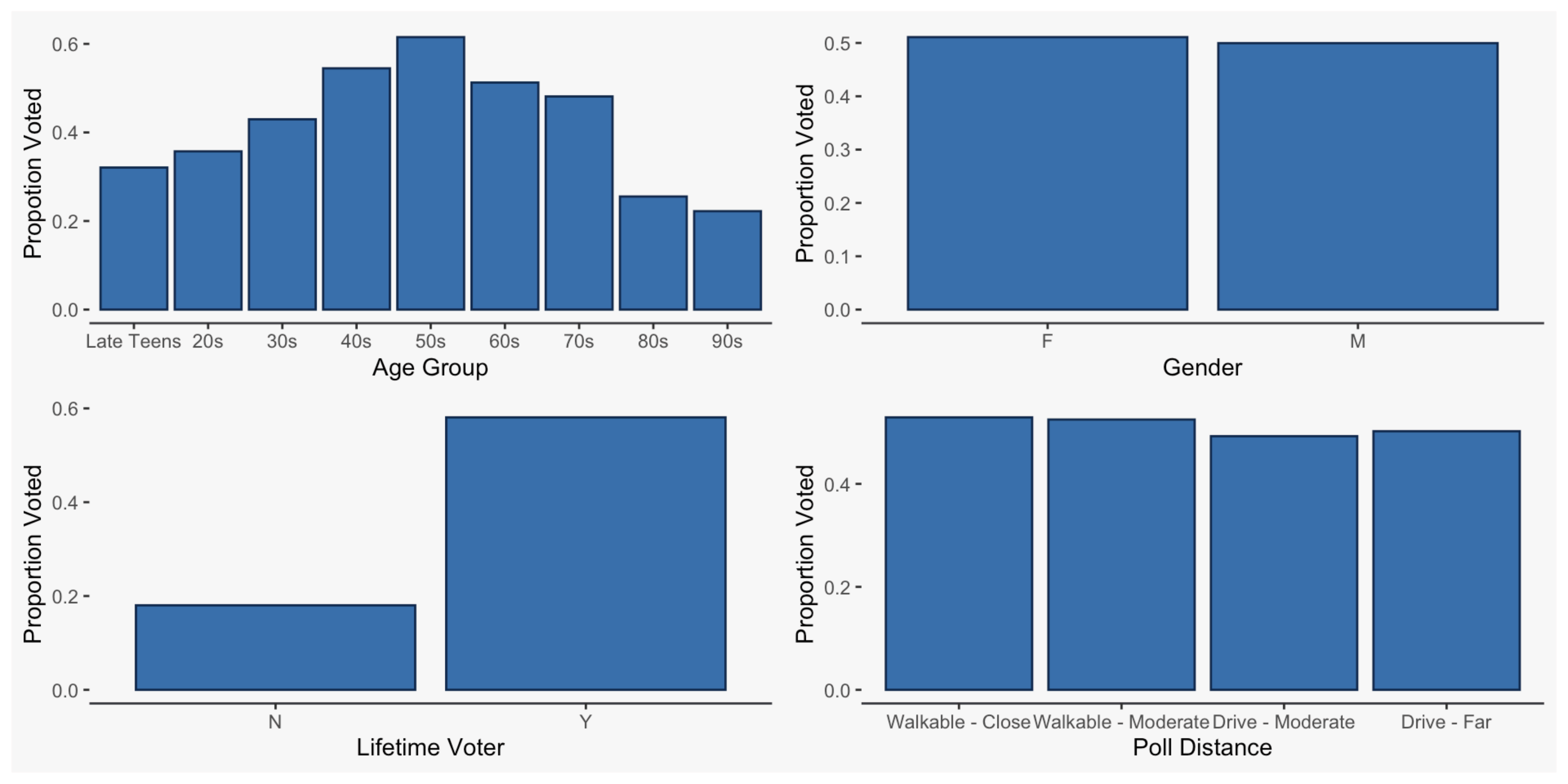
- Did they vote in the previous election [Outcome]
- Age
- Gender
- Are they a lifetime voter (voted in multiple previous elections)
- Proximity to voting location

Look at Our Data

First, let’s look at our dataset:

```
# A tibble: 5,000 × 7
   vote  gender   age age_std life_vote vote_dist          vote_dist_num
   <chr> <chr>  <int>   <dbl> <chr>     <chr>                      <int>
 1 Y     M         71  1.79   N         Walkable - Moderate            1
 2 Y     M         46 -0.0243 Y         Drive - Moderate               2
 3 Y     M         48  0.120  Y         Walkable - Moderate            1
 4 N     M         62  1.13   Y         Walkable - Moderate            1
 5 Y     M         51  0.338  Y         Drive - Moderate               2
 6 N     M         18 -2.05   Y         Drive - Moderate               2
 7 N     F         35 -0.821  Y         Drive - Far                    3
 8 N     F         59  0.917  N         Drive - Moderate               2
 9 N     F         32 -1.04   N         Drive - Moderate               2
10 N     F         26 -1.47   Y         Drive - Moderate               2
# ℹ 4,990 more rows
```

Plotting Our Data



A Main Effects Model

```r
mod_vote_me <- glm(vote ~ age_std + gender + life_vote + vote_dist_num,
                   family = binomial(link = "logit"),
                   data = data_vote)
```

	Estimate	Estimate Transf.	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.38	0.25	0.11	-12.53	0.00
age_std	0.20	1.23	0.03	6.76	0.00
genderM	-0.04	0.96	0.06	-0.75	0.45
life_voteY	1.85	6.37	0.09	20.41	0.00
vote_dist_num	-0.07	0.94	0.04	-1.81	0.07

What About Interactions?

Interactions are conditional effects—the effect of a focal predictor may change based on the value of another predictor (moderator variable):

- The effect of ability [**P**] on performance [**O**] conditional on motivation [**M**]
- The effect of a personality trait [**P**] on a behavior [**O**] conditional on the setting [**M**]
- The effect of formal planning behaviors [**P**] on new venture success conditional on the level of formal planning behaviors [**M**]

Interaction (Moderation) Jargon from Quant 1

When reading about moderation, you will likely come across several of these terms:

- **Focal Predictor**: Predictor who’s relationship with the outcome is changing because of the moderator.
- **Moderating Variable**: Predictor that is altering (moderating) the relationship between the focal predictor and outcome.
- **Conditional Effects**: The effect of the focal predictor at a specific value of the moderator.

Interactions in Our Example

We want to know if the effect of polling distance on voting changes based on whether a voter is a lifetime voter or not. Does lifetime voting status interact with polling distance to predict voting behavior?

Modeling an Interaction as a Cross-Product

To estimate an interaction effect, we create a new variable that is the product of the focal predictor and the moderator:

$$Logit = \beta_0 + \beta_1 \mathrm{PD} + \beta_2 \mathrm{LV} + \beta_3 \mathrm{PD} \times \mathrm{LV}$$

Steps to Build an Interaction Variable

Steps for building an interaction term with at least one **quantitative variable**:

1. If your **quantitative** variable does not have a meaningful 0 point, then mean center the variable.
2. If your **quantitative** variables does not have a meaningful scale, then divide it by its standard deviation.
3. Create a cross-product variable by multiplying the values of your focal predictor and moderator variable.

Building a Cross-Product Variable in Our Voting Data

```r
data_vote <-
  data_vote |>
  dplyr::mutate(
    life_vote_num = dplyr::if_else(life_vote == "Y", 1, 0),
    cp_pd_lv = vote_dist_num * life_vote_num
  )
```

```
# A tibble: 5,000 × 8
    vote gender   age life_vote vote_dist    vote_dist_num life_vote_num cp_pd_lv
   <dbl> <chr>  <int> <chr>     <chr>                <int>         <dbl>    <dbl>
 1     1 M         71 N         Walkable -…              1             0        0
 2     1 M         46 Y         Drive - Mo…              2             1        2
 3     1 M         48 Y         Walkable -…              1             1        1
 4     0 M         62 Y         Walkable -…              1             1        1
 5     1 M         51 Y         Drive - Mo…              2             1        2
 6     0 M         18 Y         Drive - Mo…              2             1        2
 7     0 F         35 Y         Drive - Far              3             1        3
 8     0 F         59 N         Drive - Mo…              2             0        0
 9     0 F         32 N         Drive - Mo…              2             0        0
10     0 F         26 Y         Drive - Mo…              2             1        2
# ℹ 4,990 more rows
```

Steps to Build a Conditional Effects (Moderation/Interaction) Model

1. Include the variables from which the cross-product is created.
2. Include the cross-product term.
3. Interpret the significance of the cross-product term to determine if an interaction exists.
4. If an interaction exists, then interpret the effects of your focal predictor conditional on specific values of the moderator.
5. **Plot your interaction to help with interpretation!**

Building a Conditional Effects (Moderation/Interaction) Model

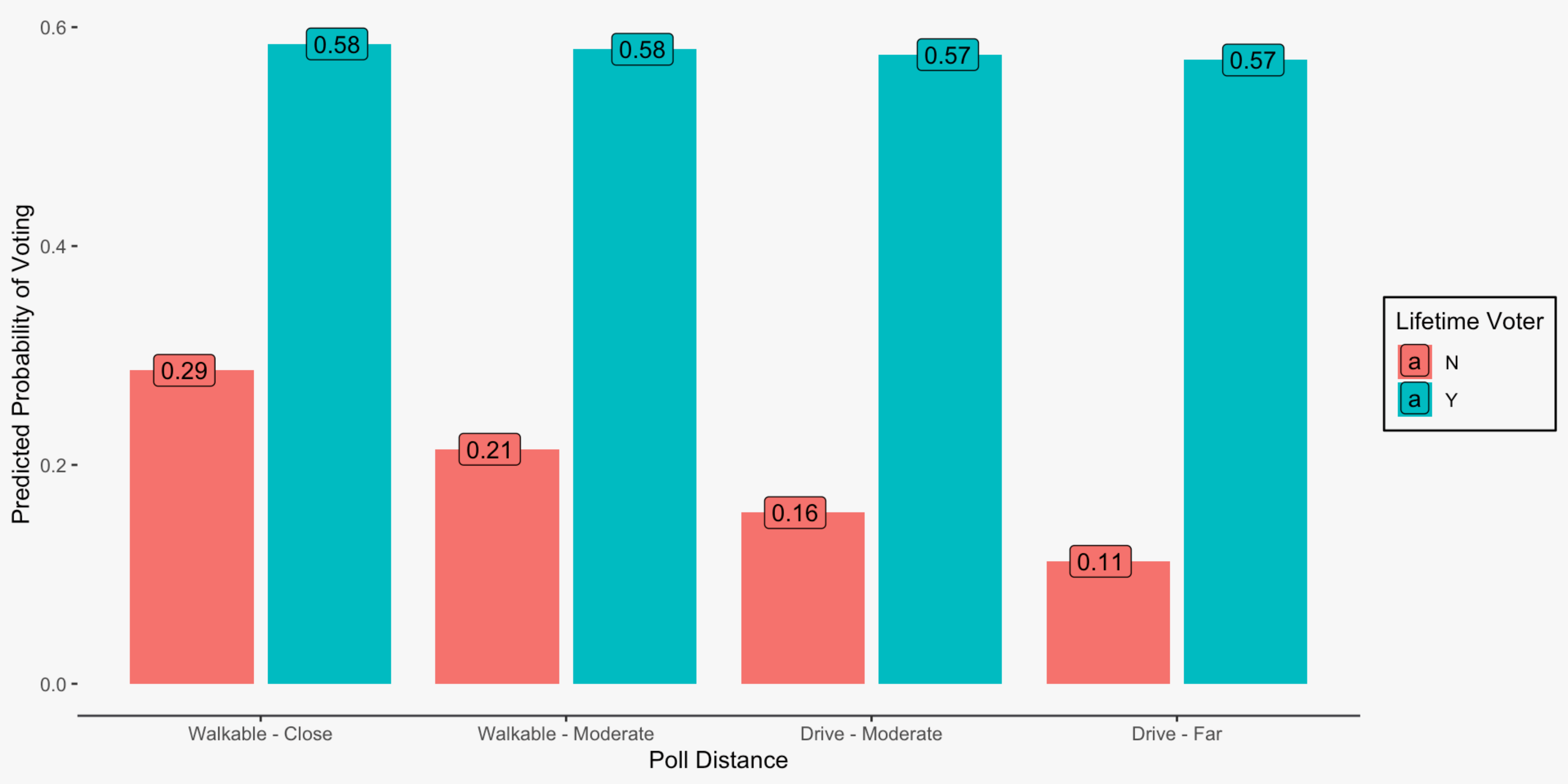
```r
mod_vote_ce <- glm(vote ~ age_std + gender + life_vote + vote_dist_num + cp_pd
                   family = binomial(link = "logit"),
                   data = data_vote)
```

	Estimate	Estimate Transf.	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.87	0.42	0.18	-4.72	0.00
age_std	0.20	1.23	0.03	6.74	0.00
genderM	-0.04	0.96	0.06	-0.75	0.46
life_voteY	1.25	3.51	0.20	6.41	0.00
vote_dist_num	-0.39	0.68	0.10	-3.76	0.00
cp_pd_lv	0.37	1.44	0.11	3.33	0.00

Checking the Significance of the Interaction

Is the z-value (test statistic) of the cross-product term larger than 1.96? If yes, then the interaction is significant.

Plotting the Interaction



Interpreting the Interaction (the Difficult Part)

Because the effects of the predictor change based on the levels of the moderator, we have to calculate the odds ratio at different levels of the moderator to get a clear understanding of the effect of the predictor.

$$\mathrm{OR} = \exp(\beta_1 + \beta_2 \times Z)$$

Interpreting the Interaction (the Difficult Part)

For lifetime voters, the odds of voting stay roughly the same regardless of polling distance, adjusting for gender and age—the odds decrease by ~2% for evey unit increase in polling distance. For non-lifetime voters, the odds of voting decrease by ~32% for each unit increase in polling distance, adjusting for gender and age.

```r
# Calculating conditional effect of polling distance for Lifetime Voter = 1
(-.386 + .366 * 1) |> exp()

# Calculating conditional effect of polling distance for Lifetime Voter = 0
(-.386 + .366 * 0) |> exp()
```

```
[1] 0.98
```

```
[1] 0.68
```

Modeling a Nonlinear Effect as a Cross-Product

To estimate the nonlinear effect of a predictor, you need to create a new variable that is the cross-product (or squared term) of the predictor with itself. Think of this as a variable interacting with itself—its effect on the outcome changes depending on the levels of the predictor variable.

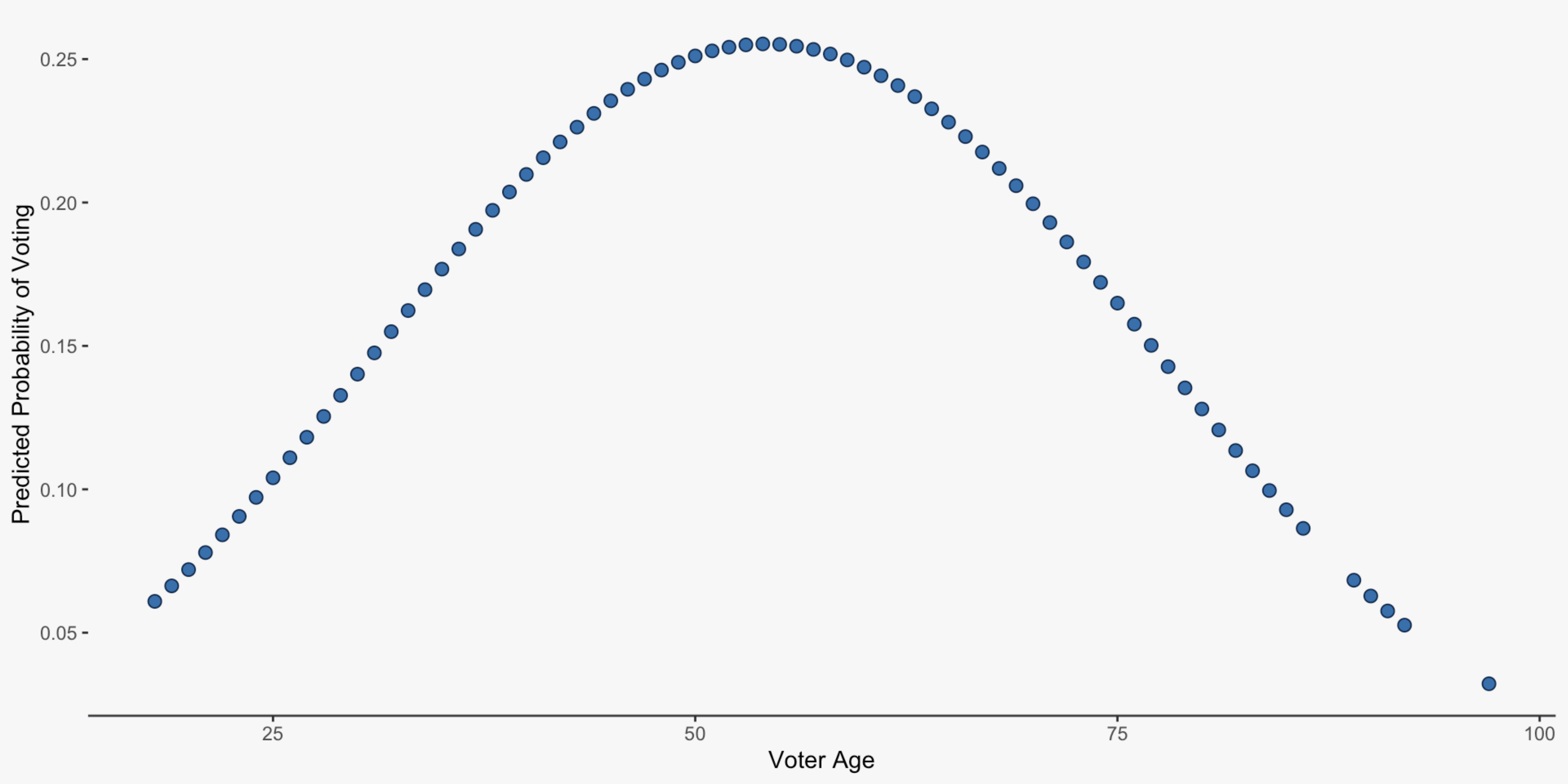
```r
data_vote <-
  data_vote |>
  dplyr::mutate(
    age_std_sq = age_std * age_std
  )
```

Building a Nonlinear Model

```r
mod_vote_nl <- glm(vote ~ age_std + age_std_sq + gender + life_vote + vote_dis
                   data = data_vote,
                   family = binomial(link = "logit"))
```

	Estimate	Estimate Transf.	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.15	0.32	0.11	-10.18	0.00
age_std	0.28	1.32	0.03	8.71	0.00
age_std_sq	-0.24	0.78	0.02	-9.99	0.00
genderM	-0.05	0.96	0.06	-0.75	0.45
life_voteY	1.87	6.50	0.09	20.50	0.00
vote_dist_num	-0.08	0.93	0.04	-2.01	0.04

Plotting Nonlinear Effects



Interpreting Nonlinear Effects

Interpreting nonlinear effects in a logistic regression model is hard! Because the odds ratio changes at different levels of the predictor, you need to calculate it at these different levels using the following formula:

$$\mathrm{OR} = \exp(\beta_1 + \beta_2 + 2 \times \beta_2 \times \boldsymbol{X})$$

Interpreting Nonlinear Effects

In the late teens (1.62 sds below the mean of 46.34), a standard deviation increase in age increases the odds of voting by 127%. At average age, a standard deviation increase in age increases the odds of voting by 5%. At older ages (1.71 sds above the mean), a standard deviation increase in age decreases the odds of voting by 54%. While adjusting for gender, lifetime voter status, and distance from the polls.

```r
# Low Value of Standardized Age: - 1.62
(.28 + -.24 + 2 * -.24 * -1.62) |> exp()

# Median Value of Standardized Age: -.02
(.28 + -.24 + -.24 * -.02) |> exp()

# High Value of Standardized Age: 1.71
(.28 + -.24 + 2 * -.24 * 1.71) |> exp()
```

```
[1] 2.27
[1] 1.05
[1] 0.46
```

Model Building with Logistic Regression

Model building is more art than science.

You decide which predictors to include or drop from your logistic regression model based on a combination of theory and statistical evidence.

You should use your theory to decide the universe of variables you are interested in and statistical evidence to determine which of those variables truly have an effect on your outcome variable.

Algorithmic Selection vs Purposeful Selection

Once you have decided on the universe of variables you are interested in, there are two broad ways you can use statistical models to determine which of those variables are significant:

- Algorithmic Selection like stepwise variable selection
- Purposeful Selection: A systematic and thoughtful approach to variable selection

Algorithmic selection can result in a final model that does not make a lot of sense, so we will focus on purposeful selection.

Steps to Purposeful Selection

Do this for each outcome variable you are investigating:

1. List all the hypotheses you have that relate to that outcome variable.
2. Build a model that includes all the predictor variables and interactions needed to test each hypothesis and all the necessary control variables.
3. If you have any non-significant interaction and/or nonlinear variables, drop them.
4. Drop any non-significant main effects that are not a part of significant interactions or nonlinear variables. This is your final model that tells you which hypotheses were supported.
5. Follow-up with model diagnostics to ensure your model fits your data well enough.

A Note on Purposeful Selection

Remember, model building is more art than science! The steps on the last slide should serve as a guide for you, especially early on in your model building career, but as you get more experience building models, you can deviate from these steps.

Pruposeful Selection with Our Models

Our hypotheses:

H1. Age is nonlinearly related to voting behavior such that there is a positive relationship between age and voting behavior when voters are younger and a negative relationship when voters are older.

H2. For first time voters, there is a negative relationship between poll distance and voting behavior such that the further away their voting poll is, the less likely they are to vote. For lifetime voters, there is no relationship between poll distance and voting behavior.

Model Building with Purposeful Selection

```r
# Build a model that contains all of our hypothesized effects and controls
mod_1 <- glm(vote ~ gender + age_std + age_std_sq + vote_dist_num * life_vote,
             data = data_vote, family = binomial(link = "logit"))
```

```
Call:
glm(formula = vote ~ gender + age_std + age_std_sq + vote_dist_num *
    life_vote, family = binomial(link = "logit"), data = data_vote)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -0.67473    0.18443  -3.659 0.000254 ***
genderM                 -0.04511    0.06085  -0.741 0.458544
age_std                  0.27467    0.03165   8.679  < 2e-16 ***
age_std_sq              -0.24115    0.02430  -9.924  < 2e-16 ***
vote_dist_num           -0.36986    0.10226  -3.617 0.000298 ***
life_voteY               1.31860    0.19607   6.725 1.75e-11 ***
vote_dist_num:life_voteY  0.33981    0.10980   3.095 0.001970 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

Model Building with Purposeful Selection

```r
# If we had non-significant interactions or nonlinear terms we would drop them

# Drop any non-significant main effects that are not a part of significant
# interactions.
mod_2 <- glm(vote ~ age_std + age_std_sq + vote_dist_num * life_vote,
             data = data_vote, family = binomial(link = "logit"))

# Use your model to determine which of your hypotheses are supported.
```

```
Call:
glm(formula = vote ~ age_std + age_std_sq + vote_dist_num * life_vote,
    family = binomial(link = "logit"), data = data_vote)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)             -0.69941    0.18142  -3.855 0.000116 ***
age_std                  0.27490    0.03165   8.687  < 2e-16 ***
age_std_sq              -0.24114    0.02430  -9.925  < 2e-16 ***
vote_dist_num           -0.36981    0.10227  -3.616 0.000299 ***
life_voteY               1.31806    0.19607   6.722 1.79e-11 ***
vote_dist_num:life_voteY 0.34016    0.10981   3.098 0.001950 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```