Skill Task 4
Regression

PS 811: Statistical Computing

Due March 6, 2020

Using the CAFE data, estimate the proportion of each party caucus that voted Yea on the auto emissions bill with regression. Set up your data by creating a binary `rep_caucus` variable that equals 1 for the Republican Caucus and 0 otherwise, and a binary `yea_vote` variable that gets 1 for Yeas and 0 for Nays (like we did for Skills Task 3).

1. Use `group_by()` and `summarize()` to estimate the proportion of `Yeas` in each party caucus, similar to Skills Task 2. This gets you a point estimate only. Your result should look like the following:

```
## # A tibble: 2 x 2
##   rep_caucus prop_yea
##        <dbl>    <dbl>
## 1          0    0.373
## 2          1    0.878
```

2. Use `lm()` to estimate a regression model where `yea_vote` is predicted by `rep_caucus`,

$$\texttt{yea\_vote}_i = \alpha + \beta(\texttt{rep\_caucus}_i) + \varepsilon_i \tag{1}$$

where i indexes Senators, α is the intercept, and β is the coefficient for being in the Republican caucus. Create a data frame of coefficients using `broom::tidy()`. Your results should appear as below. What do you notice about the coefficients compared to the proportions estimated in question 1?

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic       p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>         <dbl>    <dbl>     <dbl>
## 1 (Intercept)    0.373    0.0586      6.35 0.00000000665    0.256     0.489
## 2 rep_caucus     0.505    0.0838      6.03 0.0000000293     0.339     0.671
```

3. Generate predicted values for a generic Republican and Democratic Caucus member. Use `broom::augment()` to generate the predictions, supplying a new dataset to the

newdata argument. It is sufficient for the data you supply to newdata to contain two rows only: one Democrat (rep_caucus is 0) and one Republican (rep_caucus is 1). Results should appear as below. What do you notice about these predictions vs. the proportions estimated in question 1?

```
## # A tibble: 2 x 3
##   rep_caucus .fitted .se.fit
##        <dbl>   <dbl>   <dbl>
## 1          0   0.373  0.0586
## 2          1   0.878  0.0598
```

4. **BONUS:** Generate your predicted values with 95 percent uncertainty intervals. The notes describe how to make uncertainty intervals from augment(). Since there are fewer than 100 observations total, don't use the Normal critical value (1.96). Instead, use the appropriate *t*-value, since this is coming from an OLS model. How do you do that? Supposing that my simple model were called ols, I would find the critical value of *t* such that the CDF at *t* is 0.975, with the degrees of freedom equal to the residual degrees of freedom from my estimated model.

```
# "qt" as in, quantile function of the t distribution
critical_t <- qt(p = .975, df = glance(ols)$df.residual)
```

5. **SUPER BONUS:** When using a linear model to measure the different in means, we should probably be using heteroskedasticity-consistent errors. Try the above tasks with robust standard errors instead of a conventional standard error, and then plot your estimates and confidence intervals using geom_pointrange(). You will encounter errors that begin when you try to augment() a model with robust errors. What do you do???