title: Jan 19th in class problems author: Alex Przybycin

With the data loaded, answer the following questions. The objective here is twofold: 1) to practice your statistical computing skills, and 2) apply and explore error from fitting models on different sets of data.

1. Using some of the techniques we covered last week:
   1. Select only the Obama feeling thermometer (ftobama), household income (faminc), party affiliation on a 3 point scale (pid3), birth year (birthyr), and gender (gender) (*be sure to recode missing values to NA and omit these*)
   2. Split the subset data into training (75%) and testing (25%) sets (*hint*: remember to set the seed (set.seed()) prior to creating the split, as the proportions are generated at random)
   3. Plot the distributions of each against each other to ensure they look similar

library(here) library(tidyverse) NESdta <- read\_csv(here(“data”, “anes\_2016.csv”))

NESdta\_short <- NESdta %>% select(ftobama, pid3, birthyr, gender, faminc) head(NESdta\_short, n = 5)

set.seed(1234)

split\_tidy <- initial\_split(NESdta\_short, prop = 0.75)

train\_tidy <- training(split\_tidy) test\_tidy <- testing(split\_tidy)

tidy <- train\_tidy %>% ggplot(aes(x = ftobama)) + geom\_line(stat = “density”) + geom\_line(data = test\_tidy, stat = “density”, col = “red”) + labs(title = “Via Tidymodels”) + theme\_minimal() ```

1. Fit a linear regression (lm()) on the *training* data, predicting trump approval as a function of all other features.

training\_regression <- lm( ftobama ~ pid3+birthyr+gender+faminc, data = train\_tidy)

1. Calculate the training mean squared error (*hint*: consider using the mse() function from Dr. Soltoff’s rcfss package, which is at the uc-cfss github, *not* on CRAN).

library(rcfss)

mse(train\_tidy)

1. Calculate predictions for the testing set, using the model you built on the training set (*hint*: consider either predict() from base R, or augment() from broom).

library(base)

test\_predict = predict(train\_tidy)

1. Calculate the testing mean squared error.

training\_mse = mean(train\_tidy$residuals^2)

1. Compare the mean squared error from both sets numerically, side-by-side. What do you see? *Discuss in your groups and record a few sentences as a response.*

I’m not sure as to how to do this numerically. The function I used about gets the mean of the sum of squares from the training data.

1. Write your own function to calculate the MSE. Then, use it to re-answer questions 3 and 5. Present the results here, and compare with the rcfss approach via mse(). These results should be identical to the mse() version. Are they? If not, *why* do you think? (*just a sentence or two on your general thoughts if they differ*)

I couldn’t find the mse() function on the cfss github. Am I supposed to install that as a package in R first and find the documentation for it through R?