Gov 2018 Lab 2: Prediction with regression

Adeline Lo

February 1, 2022

Please upload your lab *with comments*, in .rmd *and* knitted forms on Gradescope so we can verify it runs without problem. For any problems that require calculations, please show your work. Finally, when doing simulations or draws from a random distribution in `R`, please set your seed immediately using `set.seed(12020)`.

This assignment will analyze vote returns for California House elections and vote choice in a presidential election.

2006 California Congressional Election Results

Our goal in this exercise is to calculate the probability that a Democratic candidate wins a House seat in a "swing district": one where the support for Democratic and Republican candidates is about equal and the incumbent is a Democrat.

Question 1

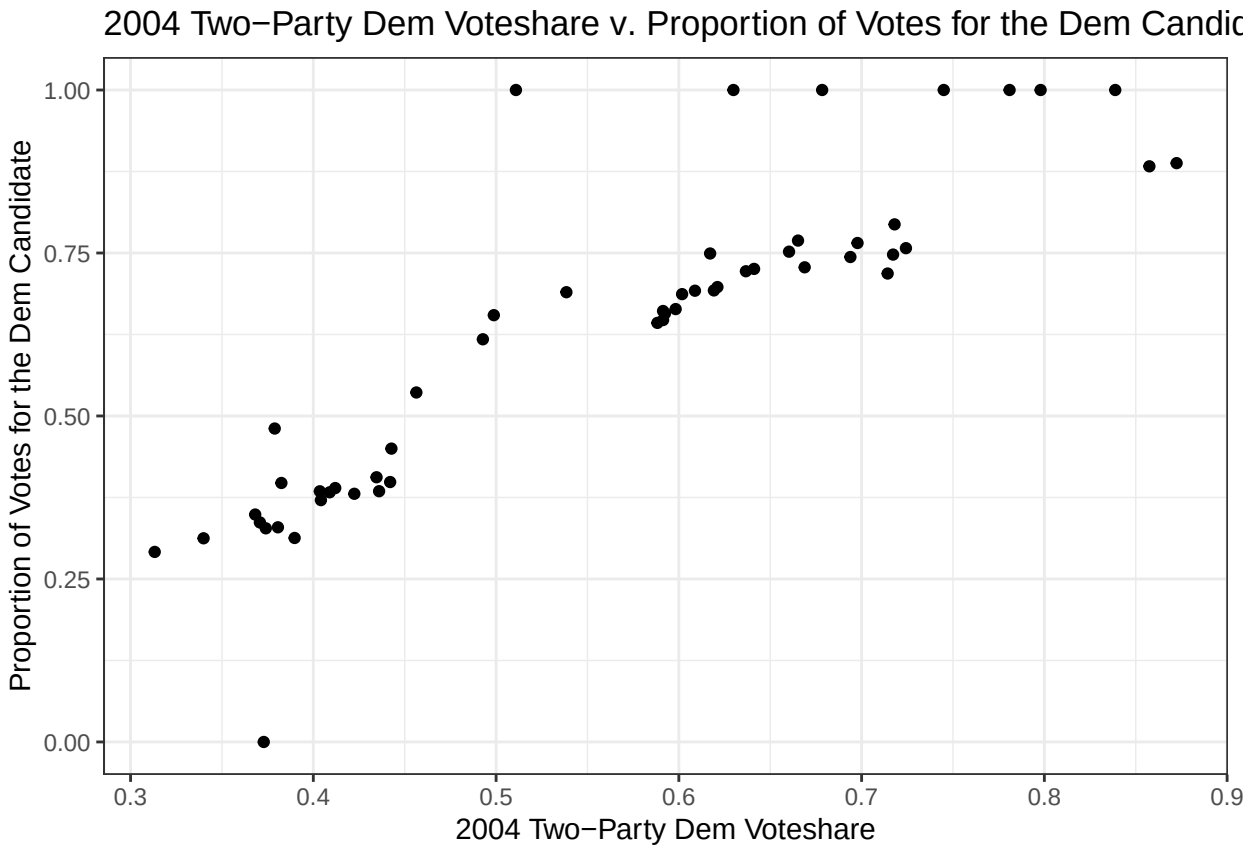
Load the data set `ca2006.csv` from your computer into R. `ca2006` a slightly modified version of the 2006 House election return data from the PSCL library.

Create a plot of the proportion of votes for the Democratic candidate, against the proportion of the two-party vote for the Democratic presidential candidate in 2004 (John Kerry) in the district. Be sure to clearly label the axes and provide an informative title for the plot

The data set contains the following variables:

- `district`: California Congressional district
- `prop_d`: proportion of votes for the Democratic candidate
- `dem_pres_2004`: proportion of two-party presidential vote for Democratic candidate in 2004 in Congressional district
- `dem_pres_2000`: proportion of two-party presidential vote for Democratic candidate in 2000 in Congressional district
- `dem_inc`: An indicator equal to 1 if the Democrat is the incumbent
- `contested`: An indicator equal to 1 if the election is contested

```
# Reading in the data
data <- read_csv("ca2006.csv")

# Plotting data
data %>%
  ggplot(aes(x = dem_pres_2004, y = prop_d)) +
  geom_point() +
  theme_bw() +
  labs(x = "2004 Two-Party Dem Voteshare",
       y = "Proportion of Votes for the Dem Candidate",
       title = "2004 Two-Party Dem Voteshare v. Proportion of Votes for the Dem Candidate")
```



Question 2

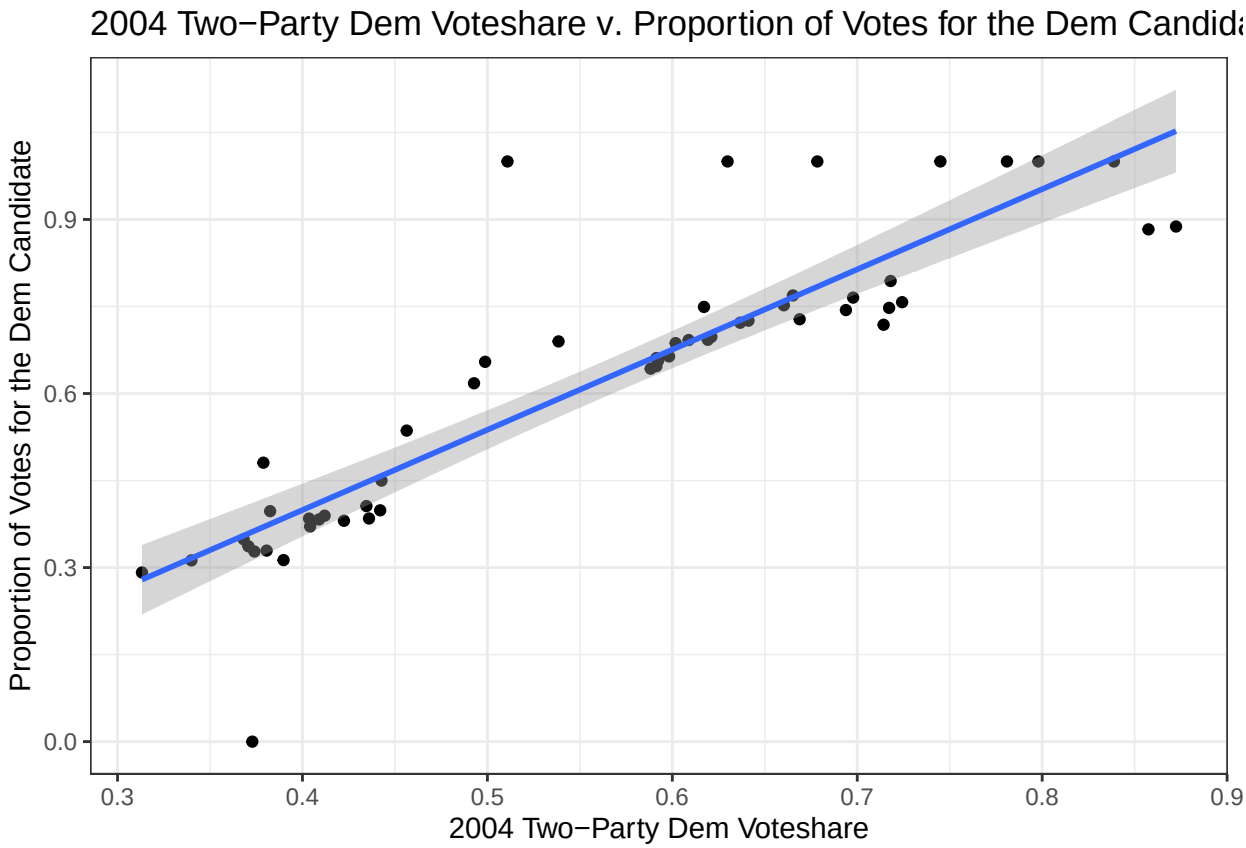
Regress the proportion of votes for the Democratic candidate, against the proportion of the two-party vote for the Democratic presidential candidate in 2004 in the district. Print the results and add the bivariate regression line to the plot.

```
# Creating model
model1 <- lm(prop_d ~ dem_pres_2004, data = data)

# Printing summary
summary(model1)
```

```
##
## Call:
## lm(formula = prop_d ~ dem_pres_2004, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36168 -0.04314 -0.00830  0.01233  0.44754
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.15390    0.05978  -2.574    0.013 *
## dem_pres_2004  1.38268    0.10291  13.436   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1125 on 51 degrees of freedom
## Multiple R-squared:  0.7797, Adjusted R-squared:  0.7754
## F-statistic: 180.5 on 1 and 51 DF,  p-value: < 2.2e-16
```

```r
# Plotting again with geom_smooth to add the bivariate regression line
data %>%
  ggplot(aes(x = dem_pres_2004, y = prop_d)) +
  geom_point() +
  theme_bw() +
  labs(x = "2004 Two-Party Dem Voteshare",
       y = "Proportion of Votes for the Dem Candidate",
       title = "2004 Two-Party Dem Voteshare v. Proportion of Votes for the Dem Candidate") +
  geom_smooth(method = "lm")
```



Question 3

Using the bivariate regression and a function you have written yourself (not `predict`!), report the expected vote share for the Democratic candidate if `dem_pres_2004` = 0.5.

```r
# Creating a function that takes in a model and one parameter value for bivariate predictions
prediction_bivariate  <- function(mod, num) {

  #Calculating prediction by multiplying parameter by coef and adding intercept
  pred <- (mod$coefficients[1] + (num * mod$coefficients[2]))
  return(pred)
}

# Outputting results
```

```
prediction_bivariate(model1, 0.5)
```

```
## (Intercept)
##   0.5374445
```

Question 4

Now, regress `prop_d` on `dem_pres_2004`, `dem_pres_2000` and `dem_inc`.

Using the multivariate regression and a function you have written yourself, report the expected vote share for the Democratic candidate if:

- `dem_pres_2004` = 0.5
- `dem_pres_2000` = 0.5
- `dem_inc` = 1

```
# Second model with new regression terms
model2 <- lm(prop_d ~ dem_pres_2004 + dem_pres_2000 + dem_inc, data = data)


# New function for multivariate predictions
prediction_multivariate  <- function(mod, nums) {

  # Adding one to multiply the intercept by
  pred_values <- c(1, nums)

  # Failsafe if lengths don't match, from class
  if(length(pred_values) != length(mod$coefficients)) stop("!!!")

  # Returning the sum of the intercept and the coefficients * X
  return(sum(pred_values * mod$coefficients))
}

# Printing prediction
prediction_multivariate(model2, c(0.5, 0.5, 1))
```

```
## [1] 0.6147444
```

Question 5

We are often interested in characterizing the uncertainty in our estimates. One nonparametric approach to estimating uncertainty is called the bootstrap. Here, we will walk through the steps to produce a function that will produce a bootstrap distribution, from which you can find the bootstrap estimates (and 95% confidence bands).

Do the following 1000 times (in a for loop; we'll learn how to vectorize/parallelize later):

a) Randomly sample 53 rows – the number of districts in California in 2006 – with replacement (set the seed). This is your bootstrap sample.
b) Using the bootstrap sample from a), fit the bivariate and multivariate regressions.
c) Using the regressions from b), predict the expected vote share for the Democratic candidate from both regressions, using the values and functions from 3) and 4)
d) Store the predictions from both regressions.
e) Repeat a-d 1000 times.

```
#Setting seed
set.seed(12020)
```

```r
# Bootstrap function
bootstrap  <- function(data) {

  # To store preds outside of loop
  bi_preds <- c()
  multi_preds <- c()

  for(i in 0:1000) {

      # Sample entire dataset with replacement
      samp <- sample_n(data, 53, replace = TRUE)

      # Creating the two model types with the sample
      bi <- lm(prop_d ~ dem_pres_2004, data = samp)
      multi <-  lm(prop_d ~ dem_pres_2004 + dem_pres_2000 + dem_inc, data = samp)

      # Making predictions using earlier functions
      bi_pred <- prediction_bivariate(bi, 0.5)
      multi_pred <- prediction_multivariate(multi, c(0.5, 0.5, 1))

      # Adding preds to list
      bi_preds <- c(bi_preds, bi_pred)
      multi_preds <- c(multi_preds, multi_pred)

  }

  # Returning a tibble of predictions for both models
  return(as_tibble(cbind(bi_preds, multi_preds)))

}

predictions <- bootstrap(data)
```

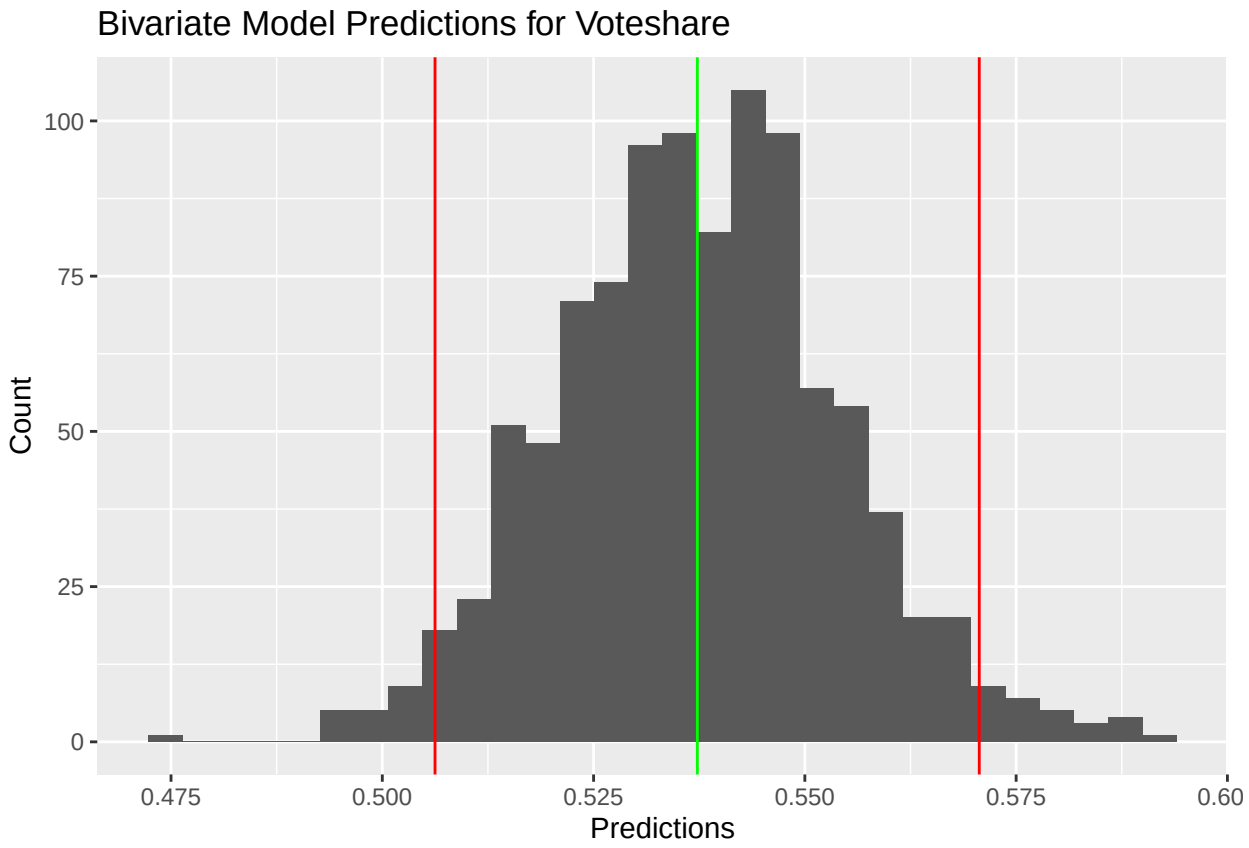
Question 6

Create histograms for both predictions adding lines for the prediction (each from Question 3 and 4) and 95% confidence bands.

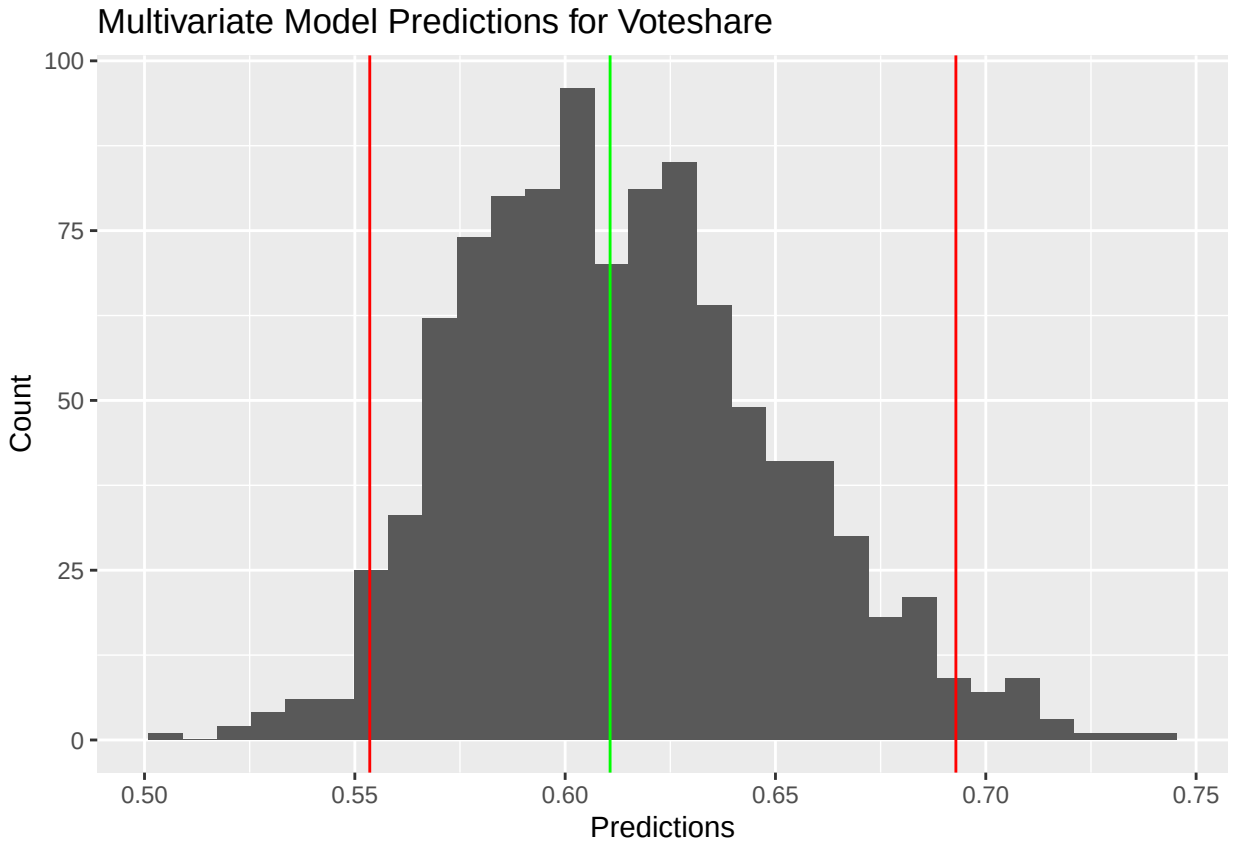
We will say the model predicts that the incumbent wins if the predicted vote share is greater than 50%. What proportion of time does each model above predict the incumbent will win?

```r
# 95% Conf ints for bivariate
bi_high <- quantile(predictions$bi_preds,.975)
bi_mid <- quantile(predictions$bi_preds,.5)
bi_low <- quantile(predictions$bi_preds,.025)


# 95% Conf ints for multivariate
multi_high <- quantile(predictions$multi_preds,.975)
multi_mid <- quantile(predictions$multi_preds,.5)
multi_low <- quantile(predictions$multi_preds,.025)

# Plotting bivariate histogram
predictions %>%
```

```
  select(bi_preds) %>%
  ggplot() +
  geom_histogram(aes(x = bi_preds)) +
  geom_vline(xintercept = bi_high, color = "red") +
  geom_vline(xintercept = bi_mid, color = "green") +
  geom_vline(xintercept = bi_low, color = "red") +
  labs(x = "Predictions",
       y = "Count",
       title = "Bivariate Model Predictions for Voteshare")
```




```
# Plotting multivariate histogram
predictions %>%
  select(multi_preds) %>%
  ggplot() +
  geom_histogram(aes(x = multi_preds)) +
  geom_vline(xintercept = multi_high, color = "red") +
  geom_vline(xintercept = multi_mid, color = "green") +
  geom_vline(xintercept = multi_low, color = "red") +
  labs(x = "Predictions",
       y = "Count",
       title = "Multivariate Model Predictions for Voteshare")
```




```
round(sum(predictions$bi_preds > 0.5) / 1000, 2)
```

```
## [1] 0.99
```

```
round(sum(predictions$multi_preds > 0.5) / 1000, 2)
```

```
## [1] 1
```

99% of predictions were greater than 0.5 for the bivariate regression, and 100% were greater for the multivariate regression.