Final Project

2025-05-18

Research Question

Chile's rapid economic growth from the 1980s through the early 2000s is often referred to as one of Latin America's economic miracles. Chile is currently a high-income country with a GNI per capita of ($15,800 current USD), higher than Mexico ($11,980) and China ($13,390) as of 2023 (World Bank, 2023). Over 60% of people lived below the poverty line in the late 1980s, but by 2022, that number fell to 5% (World Bank, 2024). Despite its economic success, Chile remains deeply unequal, a fact exposed by mass protests in 2019 after a four-percent metro fare hike (Edwards, 2023). On October 25, 2019, around 1.2 million Chileans gathered in Santiago to protest (Vergara & Luna, 2019). Though the increase was just thirty pesos (about four U.S. cents), it sparked the largest protest in Chile's democratic history and raised a pressing question: how did a booming, high-income nation celebrated as a successful economic model, face such widespread intense public discontent over a relatively minor policy change?

Mass mobilization can reflect strong civic engagement but may also signal a disconnect between citizens and the state, especially during periods of economic growth. We use 2023 Latinobarómetro data, a regional survey conducted across 18 Latin American countries, which includes a nationally representative sample of about 1,200 Chileans and covers political, economic, and social topics.

Data Cleaning

To examine protest participation, we identified survey questions related to democracy, trust, freedom, and economic outlook. Since the 2019 protests were driven by frustration over inequality, high living costs, and perceived corruption among elites, we hypothesized that these factors would be key to understanding who is more likely to protest. Because these questions use different response scales, we standardized them and generated a score between 0 and 1 for each participant in each category. An example question might include: "In general, would you say you are very satisfied, quite satisfied, not very satisfied, or not at all satisfied with the working of democracy in Chile?"

While the survey does not include a direct question about protest attendance, we created a proxy binary variable to reflect protest participation. This was based on responses to protest-related questions such as "Please tell me whether you strongly agree, agree, disagree, or strongly disagree with the following statement: Protests." We used responses from eight such questions to construct a protest engagement score, which we then converted into a binary outcome (1 = likely protester, 0 = unlikely protester) using the median as a threshold.

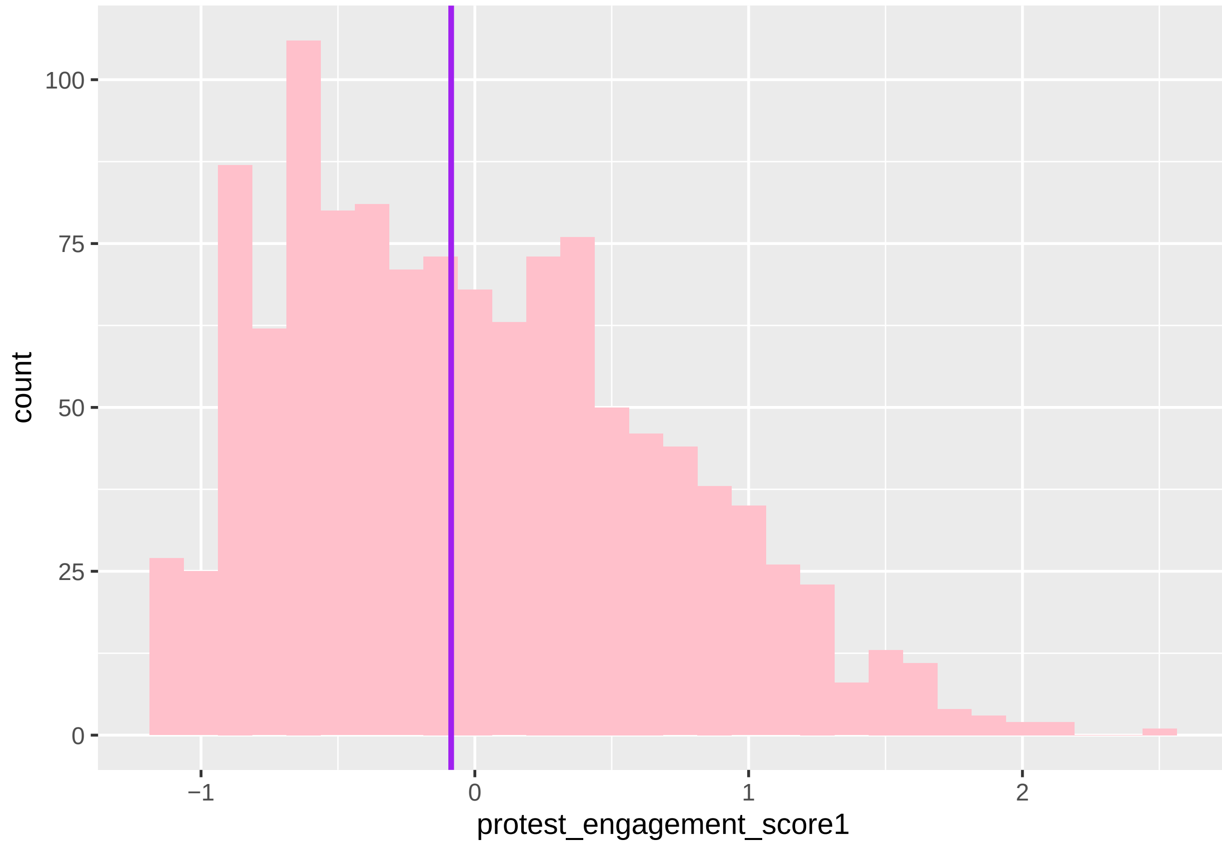
Protest Variable

```r
renamed_full <- read_dta("~/Downloads/2023_renamed.dta")
renamed_full <- renamed_full |>
  filter(idenpa == 152) |>
  mutate(across(everything(), ~ as.numeric(as.character(.))))
```

```r
#Building Protest Variable
predictors <- c(
  "talkoftenpolitics",
  "interestinpolitics",
  "workforcommunity",
  "authdemonst",
  "nonauthdemonst",
  "protestinsocmedia",
  "opinionplatform",
  "protestagree"
)

#Since 'Don't know' and 'No answer' are assigned their own values in the dataset, we recoded them as NA
renamed_full_clean <-renamed_full  |>
  mutate(across(
    all_of(predictors),
    ~ if_else(.x %in% c(8, 97, 98, 99, -5, -4, -3, -1, -2), NA_real_, as.numeric(.x))
  ))

#Standardize
protest_scaled <- scale(renamed_full_clean[, predictors])
protest_scaled_df <- as.data.frame(protest_scaled)

#Reverse the order (higher values = more positive attitudes)
protest_scaled_sd <- protest_scaled_df |>
  mutate(protest_engagement_score1 = -rowMeans(across(all_of(predictors)), na.rm = TRUE))

#Scaling
renamed_full_clean <- renamed_full_clean |>
  mutate(
    protest_engagement_score1 = -rowMeans(scale(across(all_of(predictors))), na.rm = TRUE)
  )

#Look at the distribution of the protest engagement variable as well as the median
renamed_full_clean |>
  ggplot() +
  geom_histogram(aes(x = protest_engagement_score1), fill = "pink") +
  geom_vline(
    xintercept = median(protest_scaled_sd$protest_engagement_score1, na.rm = TRUE),
    color = "purple",
    linewidth = 1
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
threshold <- median(renamed_full_clean$protest_engagement_score1, na.rm = TRUE)

renamed_full_cleann <- renamed_full_clean |>
  mutate(protest_attend = if_else(protest_engagement_score1 > threshold, 1, 0))
```

The histogram shows a right-skewed distribution, with most respondents clustered near 0 and a long tail of higher protest engagement scores. This suggests that while many are only modestly engaged, a small but notable group is highly protest-prone. To create a binary protest variable, we use the median as the threshold, since it's more reliable than the mean in a skewed distribution. Scores below the median are coded as 0 (low engagement), and scores at or above the median are coded as 1 (high engagement).

Democracy, Trust, Economy, and Freedom Indices

```
#Choosing the relevant survey questions for each category
democracy_vars <- c(
  "democracypreference", "democracysatisfaction",
  "governedbyfew", "democracybest")

trust_vars <- c(
  "trustinarmedforces", "trustinpolice", "trustinchurch", "trustincongress",
  "trustinnationalgovernment", "trustinjudiciary", "trustinpoliticalparties",
  "trustinelectoralinst", "trustinnatcompanies", "trustintradeunions",
  "trustintelevision", "trustintlcompanies", "trustinbanks", "trustinradio",
  "trustinprintedpress", "trustinsocmedia", "presleadershipapproval",
```

```r
  "trustinpeople"
)

economic_vars <- c(
  "lifesatisfaction", "countryprogress", "countryproblems",
  "countryeconomicsituation", "countrypast12monthseconsituation",
  "countrynext12monthseconsituation", "familynext12monthseconsituation",
  "incomedistfairness", "economysatisfaction"
)

freedom_vars <- c(
  "freedompoliticalpart", "freedomenv", "freedomprivateproperty",
  "justfairwealthdistf", "equalityofmenwomenf", "chanceswuoriginf",
  "freedomofspeech", "freedomofreligion", "protectionagainstcrimef",
  "socialsecurityf", "freedomchancetogetjob", "publicopinionexpression"
)

#more cleaning
renamed_full_cleannp <- renamed_full_cleann |>
  mutate(across(
    all_of(c(democracy_vars, trust_vars, economic_vars, freedom_vars)),
    ~ if_else(.x %in% c(8, 97, 98, 99, -5, -4, -3, -1, -2), NA_real_, as.numeric(.x))
  ))

renamed_full_scaled <- renamed_full_cleannp %>%
  mutate(
    democracy_score = rowMeans(select(., all_of(democracy_vars)), na.rm = TRUE),
    trust_score     = rowMeans(select(., all_of(trust_vars)), na.rm = TRUE),
    economic_score  = rowMeans(select(., all_of(economic_vars)), na.rm = TRUE),
    freedom_score   = rowMeans(select(., all_of(freedom_vars)), na.rm = TRUE)
  )

renamed_full_cleaner <- renamed_full_scaled  |>
  mutate(
    democracy_new = (democracy_score - min(democracy_score, na.rm = TRUE)) /
                            (max(democracy_score, na.rm = TRUE) - min(democracy_score, na.rm = TRUE)),

    trust_new = (trust_score - min(trust_score, na.rm = TRUE)) /
                        (max(trust_score, na.rm = TRUE) - min(trust_score, na.rm = TRUE)),

    economic_new = (economic_score - min(economic_score, na.rm = TRUE)) /
                           (max(economic_score, na.rm = TRUE) - min(economic_score, na.rm = TRUE)),

    freedom_new = (freedom_score - min(freedom_score, na.rm = TRUE)) /
                          (max(freedom_score, na.rm = TRUE) - min(freedom_score, na.rm = TRUE))
  )
```

Demographics

Using our contextual knowledge, we filtered out relevant demographic variables in addition to the four categories because there are over 200 variables in the dataset. We included (and cleaned) 9 variables such as age, socio-economic class, religion, devoutness, race, education, and political leaning.

```
demographics <- c("religion",
  "religiondevout",
  "socioeconclass",
  "incomeenough",
  "race",
  "education",
  "householdheademp",
  "leftrightscale")

renamed_full_cleaner11 <- renamed_full_cleaner |>
  mutate(across(
    all_of(c(demographics)),
    ~ if_else(.x %in% c(8, 97, 98, 99, -5, -4, -3, -1, -2), NA_real_, as.numeric(.x))
  ))

protestData <- renamed_full_cleaner11
```

Random Forest

Using four thematic categories and nine demographic variables, we built a random forest model to predict protest participation. This method is effective because it combines predictions from multiple decision trees, each trained on different subsets of the data. For example, one tree might weigh views on democracy and age, while another focuses on economic outlook and trust. Each tree casts a "vote," and the model aggregates these to make a final prediction. By using many trees and varying the input variables, the model avoids overfitting and improves reliability. Random forest also handles both numerical and categorical variables well.

First Stage

```
library(ranger)
```

```
##
## Attaching package: 'ranger'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```
set.seed(1)

protestData <- protestData |>
  mutate(across(c(protest_attend, religion, religiondevout, socioeconclass, incomeenough, race, educatio

protestDataClean13vars <- protestData |>
                select(protest_attend, religion, religiondevout, socioeconclass,
                       incomeenough, race, education, age, leftrightscale,
                       democracy_score, trust_score, economic_score, freedom_score) |>
  na.omit() |>
  mutate(across(c(protest_attend, religion, religiondevout, socioeconclass, incomeenough, race, educatio
```

```r
#looking at data without NAs just to see which variables are most important
rf_NAs <- ranger(protest_attend ~ . - protest_label -protest_engagement_score1,
             data = protestDataClean13vars,
             importance = "impurity")
```

```
## Warning in terms.formula(f, data = data): 'varlist' has changed (from nvar=13)
## to new 15 after EncodeVars() -- should no longer happen!
```

```r
rf_NAs
```

```
## Ranger result
##
## Call:
##  ranger(protest_attend ~ . - protest_label - protest_engagement_score1,      data = protestDataClean
##
## Type:                             Classification
## Number of trees:                  500
## Sample size:                      443
## Number of independent variables:  12
## Mtry:                             3
## Target node size:                 1
## Variable importance mode:         impurity
## Splitrule:                        gini
## OOB prediction error:             37.25 %
```

```r
rf_NAs$confusion.matrix
```

```
##     predicted
## true   0   1
##    0 141  81
##    1  84 137
```

```r
sort(rf_NAs$variable.importance)
```

```
##        religion  religiondevout    incomeenough            race       education
##        4.291473        9.176812        9.663524       10.167778       10.652751
##  socioeconclass  leftrightscale democracy_score  economic_score   freedom_score
##       11.735360       18.269559       21.634716       25.790530       27.468707
##     trust_score             age
##       33.696967       37.047855
```

```r
#distribution of 0s and 1s in protest participation
table(protestData$protest_attend)
```

```
##
##   0   1
## 600 598
```

```r
cmna <- rf_NAs$confusion.matrix

# Accuracy for predicting 0s
accuracy_0 <- cmna[1, 1] / sum(cmna[1, ])
# Accuracy for predicting 1s
accuracy_1 <- cmna[2, 2] / sum(cmna[2, ])
overall_accuracy <- sum(diag(cmna)) / sum(cmna)

print(paste("Accuracy for 0s:", round(accuracy_0 * 100, 1), "%"))
```

```
## [1] "Accuracy for 0s: 63.5 %"
```

```r
print(paste("Accuracy for 1s:", round(accuracy_1 * 100, 1), "%"))
```

```
## [1] "Accuracy for 1s: 62 %"
```

```r
print(paste("Overall Accuracy:", round(overall_accuracy * 100, 1), "%"))
```

```
## [1] "Overall Accuracy: 62.8 %"
```

The proportion of protest attendance is fairly balanced, so class imbalance isn't our concern. Our model currently has 62.8% overall accuracy, with 63.5% accuracy for predicting 0s and 62% for predicting 1s. We plan to make adjustments to improve performance.

After removing missing values, the most important predictors—ranked from highest to lowest are: trust_score, economic_score, age, freedom_score, democracy_score, socioeconclass, leftrightscale, race, education, incomeenough, religiondevout, and religion.

Next, we check for missing values in the top five predictors in protestData.

```r
colSums(is.na(protestData |>
                select(protest_attend, trust_score, age, economic_score,freedom_score, democracy_score)
```

```
##  protest_attend     trust_score             age   economic_score   freedom_score
##               2               0               0                0               2
## democracy_score
##               0
```

```r
#finding the missing values rows
which(is.na(protestData$protest_attend))
```

```
## [1]  421 1073
```

```r
protestData[421, ]
```

```
## # A tibble: 1 x 284
##   numinves idenpa numentre    reg    ciudad tamciud comdist   age  sexo codigo
##      <dbl>  <dbl>    <dbl>  <dbl>     <dbl>   <dbl>   <dbl> <dbl> <dbl>  <dbl>
## 1       23    152      421 152014 152000037       8     137    27     2   3928
## # i 274 more variables: diareal <dbl>, mesreal <dbl>, ini <dbl>, fin <dbl>,
```

```
## #   dura <dbl>, totrevi <dbl>, totcuot <dbl>, totrech <dbl>, totperd <dbl>,
## #   numcasa <dbl>, codsuper <dbl>, supervvi <dbl>, superven <dbl>, codif <dbl>,
## #   digit <dbl>, lifesatisfaction <dbl>, countryprogress <dbl>, P3N <dbl>,
## #   countryproblems <dbl>, countryeconomicsituation <dbl>,
## #   countrypast12monthseconsituation <dbl>,
## #   countrynext12monthseconsituation <dbl>, ...
```

```r
protestData[1073, ]
```

```
## # A tibble: 1 x 284
##   numinves idenpa numentre    reg    ciudad tamciud comdist   age  sexo codigo
##      <dbl>  <dbl>    <dbl>  <dbl>     <dbl>   <dbl>   <dbl> <dbl> <dbl>  <dbl>
## 1       23    152     1073 152007 152009036       1     149    27     1   2300
## # i 274 more variables: diareal <dbl>, mesreal <dbl>, ini <dbl>, fin <dbl>,
## #   dura <dbl>, totrevi <dbl>, totcuot <dbl>, totrech <dbl>, totperd <dbl>,
## #   numcasa <dbl>, codsuper <dbl>, supervvi <dbl>, superven <dbl>, codif <dbl>,
## #   digit <dbl>, lifesatisfaction <dbl>, countryprogress <dbl>, P3N <dbl>,
## #   countryproblems <dbl>, countryeconomicsituation <dbl>,
## #   countrypast12monthseconsituation <dbl>,
## #   countrynext12monthseconsituation <dbl>, ...
```

Since protest_attend and freedom_score had some missing values, we used MICE to impute them. Although only a few observations were affected, imputation allowed us to avoid data loss. Understanding why data is missing is important. Here, it was likely due to skipped survey questions rather than random error, so imputation helps preserve the overall pattern without distortion.

```r
set.seed(1)
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
imputedData <- mice(protestData |> select(freedom_score, protest_attend),
                         method = "rf",
                         seed = 1)
```

```
##
##  iter imp variable
##   1   1  freedom_score  protest_attend
##   1   2  freedom_score  protest_attend
##   1   3  freedom_score  protest_attend
##   1   4  freedom_score  protest_attend
##   1   5  freedom_score  protest_attend
```

```
## 2 1 freedom_score protest_attend
## 2 2 freedom_score protest_attend
## 2 3 freedom_score protest_attend
## 2 4 freedom_score protest_attend
## 2 5 freedom_score protest_attend
## 3 1 freedom_score protest_attend
## 3 2 freedom_score protest_attend
## 3 3 freedom_score protest_attend
## 3 4 freedom_score protest_attend
## 3 5 freedom_score protest_attend
## 4 1 freedom_score protest_attend
## 4 2 freedom_score protest_attend
## 4 3 freedom_score protest_attend
## 4 4 freedom_score protest_attend
## 4 5 freedom_score protest_attend
## 5 1 freedom_score protest_attend
## 5 2 freedom_score protest_attend
## 5 3 freedom_score protest_attend
## 5 4 freedom_score protest_attend
## 5 5 freedom_score protest_attend
```

```r
completeData <- complete(imputedData)

#fill in NAs with complete data
protestData$freedom_score <- completeData$freedom_score
protestData$protest_attend <- completeData$protest_attend


#Check to see if there are still NAs
protestDataClean13vars <- protestData |>
  select(protest_attend, religion, religiondevout, socioeconclass,
                      incomeenough, race, education, age, leftrightscale,
                      democracy_score, trust_score, economic_score, freedom_score) |>
  mutate(across(c(protest_attend, religion, religiondevout, socioeconclass, incomeenough, race, educati

colSums(is.na(protestDataClean13vars |>
                select(protest_attend, freedom_score, trust_score, economic_score, democracy_score, age
```

```
##  protest_attend   freedom_score     trust_score  economic_score democracy_score
##               0               0               0               0               0
##             age
##               0
```

Now that there are no more NAs in the original dataset, we will run random forest using the top 5 explanatory variables.

```r
set.seed(1)
rf <- ranger(protest_attend ~ trust_score+age+economic_score+freedom_score+democracy_score,
             data = protestDataClean13vars,
             importance = "impurity")

rf
```

```
## Ranger result
##
## Call:
##  ranger(protest_attend ~ trust_score + age + economic_score +      freedom_score + democracy_score, d
##
## Type:                             Classification
## Number of trees:                  500
## Sample size:                      1200
## Number of independent variables:  5
## Mtry:                             2
## Target node size:                 1
## Variable importance mode:         impurity
## Splitrule:                        gini
## OOB prediction error:             34.75 %
```

```
rf$confusion.matrix
```

```
##     predicted
## true   0   1
##    0 375 227
##    1 190 408
```

```
cm1 <- rf$confusion.matrix

accuracy_0 <- cm1[1, 1] / sum(cm1[1, ])
accuracy_1 <- cm1[2, 2] / sum(cm1[2, ])
overall_accuracy <- sum(diag(cm1)) / sum(cm1)

print(paste("Accuracy for 0s:", round(accuracy_0 * 100, 1), "%"))
```

```
## [1] "Accuracy for 0s: 62.3 %"
```

```
print(paste("Accuracy for 1s:", round(accuracy_1 * 100, 1), "%"))
```

```
## [1] "Accuracy for 1s: 68.2 %"
```

```
print(paste("Overall Accuracy:", round(overall_accuracy * 100, 1), "%"))
```

```
## [1] "Overall Accuracy: 65.2 %"
```

We see that the prediction error is about the same when we ran random forest on the data that omitted the NAs. Instead of imputing data using mice, what if we did something more naive such as imputing data using mean/median?

```
set.seed(1)
#reload original protestData

protestData <- protestData |>
  mutate(across(c(protest_attend, religion, religiondevout, socioeconclass, incomeenough, race, educatio
```

```r
#select columns with NAs, find median, and update the columns in protestData
protestDataNoNA <- protestData |>
  select(freedom_score)

medianFreedomScore <- median(protestDataNoNA$freedom_score)

protestDataMedianAsNAs <- protestDataClean13vars |>
    mutate(freedom_score = case_when(is.na(freedom_score) ~ medianFreedomScore, TRUE ~ freedom_score))
  filter(!is.na(protest_attend)) #gets rid of 6 missing protest_attend NA values instead of calculating

rfMedians <- ranger(protest_attend ~ trust_score+age+economic_score+freedom_score+democracy_score,
             data = protestDataMedianAsNAs,
             importance = "impurity")

rfMedians
```

```
## Ranger result
##
## Call:
##  ranger(protest_attend ~ trust_score + age + economic_score +      freedom_score + democracy_score, d
##
## Type:                             Classification
## Number of trees:                  500
## Sample size:                      1200
## Number of independent variables:  5
## Mtry:                             2
## Target node size:                 1
## Variable importance mode:         impurity
## Splitrule:                        gini
## OOB prediction error:             34.75 %
```

```r
rfMedians$confusion.matrix
```

```
##     predicted
## true   0   1
##    0 375 227
##    1 190 408
```

Since median imputation resulted in the same prediction error (34.75%) as the model using MICE, we chose to proceed with the MICE-imputed data and split the dataset into training and testing sets for prediction.

```r
protestData <- protestData |>
  mutate(across(c(protest_attend, religion, religiondevout, socioeconclass, incomeenough, race, educatio

protestData$freedom_score <- completeData$freedom_score
protestData$protest_attend <- completeData$protest_attend

protestDataClean13vars <- protestData |>
                select(protest_attend, religion, religiondevout, socioeconclass,
                       incomeenough, race, education, age, leftrightscale,
                       democracy_score, trust_score, economic_score, freedom_score) |>
  mutate(across(c(protest_attend, religion, religiondevout, socioeconclass, incomeenough, race, educatio
```

```r
colSums(is.na(protestData |>
                select(protest_attend, freedom_score, trust_score, economic_score, democracy_score, age)
```

```
##  protest_attend   freedom_score     trust_score  economic_score democracy_score
##               0               0               0               0               0
##             age
##               0
```

```r
#training/testing
set.seed(2)
split <- sample(1:nrow(protestDataClean13vars), 0.5*nrow(protestDataClean13vars))
train <- protestDataClean13vars[split,]
test <- protestDataClean13vars[-split, ]

rfTrain <- ranger(protest_attend ~ trust_score+age+economic_score+freedom_score+democracy_score,
             data = train,
             importance = "impurity")

rfTrainPreds <- predict(rfTrain, data = test)
cm3 <- table(rfTrainPreds$predictions, test$protest_attend)

accuracy_0 <- cm3[1, 1] / sum(cm3[1, ])
accuracy_1 <- cm3[2, 2] / sum(cm3[2, ])
overall_accuracy <- sum(diag(cm3)) / sum(cm3)

print(paste("Accuracy for 0s:", round(accuracy_0 * 100, 1), "%"))
```

```
## [1] "Accuracy for 0s: 64.1 %"
```

```r
print(paste("Accuracy for 1s:", round(accuracy_1 * 100, 1), "%"))
```

```
## [1] "Accuracy for 1s: 60.8 %"
```

```r
print(paste("Overall Accuracy:", round(overall_accuracy * 100, 1), "%"))
```

```
## [1] "Overall Accuracy: 62.3 %"
```

The current model achieves 62.3% overall accuracy and does slightly better (65%) at identifying people who are unlikely to attend protests. While this is better than random guessing, it suggests that our selected 13 variables (covering attitudes toward democracy, trust, freedom, and economic outlook) capture only part of the story.

To improve the model, we decided to include the full set of available survey variables to see if other factors better predict protest participation.

Random Forest with All Variables

```r
library(ranger)

sum(is.na(protestData$protest_attend))
```

```
## [1] 0
```

```r
protestData1 <- protestData |>
  select(-all_of(predictors)) |>
   select (-protest_engagement_score1)

protestData1 <- protestData1 |>
  mutate(protest_attend = as.factor(protest_attend)) |>
  filter(!is.na(protest_attend))

rfAll <- ranger(protest_attend ~ .,
             data = protestData1,
             importance = "impurity")

sort(rfAll$variable.importance, decreasing = TRUE)[1:10]
```

```
##   P45ST_A   P44ST_C       S17       S11       age       S10    ciudad   P44ST_D
## 40.535023 22.484467 10.628843  9.188212  8.259935  7.068245  7.039201  6.553873
##     P47ST    codigo
##  6.494133  6.313527
```

```r
colSums(is.na(protestData1 |>
  select(P45ST_A, P44ST_C, economic_score, age, S14M_F, P44ST_D, S11, S12)))
```

```
##        P45ST_A        P44ST_C economic_score            age         S14M_F
##              0              0              0              0              0
##        P44ST_D            S11            S12
##              0              0              0
```

```r
#no NAs
```

Since protest_engagement_score1 and protest_label are identical to protest_attend, we removed them from the top 10 list and excluded the variables that were used to construct protest_attend. After filtering them out, we see that P45ST_A (sign a petition) and P44ST_C (talk about politics with friends) ranked highest in importance, along with economic_score, age, S14M_F (social media platform), P44ST_D (working for a political party or candidate), S11 (length of education), and S12 (length of parents education). These variables are interesting because they tell us that actions like signing a petition or working for a political party are more strongly linked to a person's likelihood of protesting. The remaining in the list are nonsensical variables.

Let's use the top 8 variables now.

Split the data into train/test

```r
set.seed(123)
sample_index <- sample(nrow(protestData1), size = 0.5 * nrow(protestData1))
train_data <- protestData1[sample_index, ]
test_data <- protestData1[-sample_index, ]
```

Using our random forest on the train/test data

```r
library(caret)

rfTopEight <- ranger(
  formula = protest_attend ~ P45ST_A + P44ST_C + economic_score + age +
                             S14M_F + P44ST_D + S11 + S12,
  data = train_data,
  importance = "impurity"
)

rfTopEight_preds <- predict(rfTopEight, data = test_data)$predictions

cm_top8 <- table(Predicted = rfTopEight_preds, Actual = test_data$protest_attend)

accuracy_0 <- cm_top8[1, 1] / sum(cm_top8[1, ])
accuracy_1 <- cm_top8[2, 2] / sum(cm_top8[2, ])
overall_accuracy <- sum(diag(cm_top8)) / sum(cm_top8)

print(paste("Accuracy for 0s:", round(accuracy_0 * 100, 1), "%"))
```

```
## [1] "Accuracy for 0s: 84.7 %"
```

```r
print(paste("Accuracy for 1s:", round(accuracy_1 * 100, 1), "%"))
```

```
## [1] "Accuracy for 1s: 72.6 %"
```

```r
print(paste("Overall Accuracy:", round(overall_accuracy * 100, 1), "%"))
```

```
## [1] "Overall Accuracy: 78.2 %"
```

The model now achieves 85.7% accuracy for predicting 0s, 74.4% for predicting 1s, and 79.67% overall. Although variables related to democracy, freedom, and trust were part of our earlier models, they did not rank among the top predictors of protest participation. This does not mean these attitudes are unimportant, but rather that they may be less directly tied to protest behavior than concrete political actions, even though economic outlook was ranked one of the most important variables. Our final model tells us that variables like signing a petition, working for a political party, age, and social media use are stronger indicators. These results suggest that direct forms of political engagement are more effective at identifying protestors than broader ideological attitudes like satisfaction with democracy.

General Linear Model/Logistic Regression

We also use Logistic regression, which is a type of Generalized Linear Model (GLM) designed for binary outcomes like whether someone attended a protest or not. In our case, it estimates the probability that a respondent participated in a protest based on predictors such as age, economic outlook, education, freedom, and democracy satisfaction. The model uses the logistic function to map input values to probabilities between 0 and 1 and helps us classify individuals as protestors or non-protestors based on their characteristics.

Now we pull back our core predictors-democracy, trust, freedom, economic outlook, and key demographics.

```
variables_of_interest <- c("religion", "religiondevout", "socioeconclass", "incomeenough","race","educa
)

colSums(is.na(
  protestData1 |>
    select(
      religion, religiondevout, socioeconclass, incomeenough,
      race, education, age, leftrightscale,
      democracy_score, trust_score, economic_score, freedom_score
    )
))
```

```
##        religion  religiondevout  socioeconclass    incomeenough            race
##             404             425               8               7             117
##       education             age  leftrightscale democracy_score     trust_score
##               0               0             420               0               0
##  economic_score   freedom_score
##               0               0
```

```
variables_of_interest1 <- c("socioeconclass", "incomeenough","race","education","age", "democracy_score
)

#impute the missing values

logit_df <- protestData1[, c(variables_of_interest1, "protest_attend")]

imputed <- mice(logit_df, method = "pmm", seed = 123)
```

```
##
##  iter imp variable
##   1   1  socioeconclass  incomeenough  race
##   1   2  socioeconclass  incomeenough  race
##   1   3  socioeconclass  incomeenough  race
##   1   4  socioeconclass  incomeenough  race
##   1   5  socioeconclass  incomeenough  race
##   2   1  socioeconclass  incomeenough  race
##   2   2  socioeconclass  incomeenough  race
##   2   3  socioeconclass  incomeenough  race
##   2   4  socioeconclass  incomeenough  race
##   2   5  socioeconclass  incomeenough  race
##   3   1  socioeconclass  incomeenough  race
##   3   2  socioeconclass  incomeenough  race
##   3   3  socioeconclass  incomeenough  race
```

```
##   3   4  socioeconclass  incomeenough  race
##   3   5  socioeconclass  incomeenough  race
##   4   1  socioeconclass  incomeenough  race
##   4   2  socioeconclass  incomeenough  race
##   4   3  socioeconclass  incomeenough  race
##   4   4  socioeconclass  incomeenough  race
##   4   5  socioeconclass  incomeenough  race
##   5   1  socioeconclass  incomeenough  race
##   5   2  socioeconclass  incomeenough  race
##   5   3  socioeconclass  incomeenough  race
##   5   4  socioeconclass  incomeenough  race
##   5   5  socioeconclass  incomeenough  race
```

```r
completed_df <- complete(imputed)
completed_df$protest_attend <- as.factor(protestData1$protest_attend)
```

Religion, religiondevout, and leftrightscale are missing for about half the sample, so we drop them rather than imputing. For the rest, we used predictive mean matching (pmm) during the imputation step to handle missing values. PMM is a method that fills in missing values by finding observed values with similar predicted means and randomly drawing from them. This approach maintains realistic data distributions and is particularly useful when the data is not normally distributed or when preserving observed values is important.

Split into test/train data

This code uses logistic regression to predict whether someone attended a protest based on key variables like age, education, and political views. We start by randomly splitting the data in half—one part to train the model, and the other to test how well it works. The model estimates how likely each person in the test set is to have protested. If that likelihood is 0.5 or higher, we classify them as a protestor. We then compare the model's predictions to the actual responses using a confusion matrix to see how accurately it identified protestors and non-protestors.

```r
set.seed(123)
sample_index <- sample(nrow(completed_df), size = 0.5 * nrow(completed_df))
train_data <- completed_df[sample_index, ]
test_data <- completed_df[-sample_index, ]

logit_model <- glm(protest_attend ~ ., data = train_data, family = "binomial")
pred_probs <- predict(logit_model, newdata = test_data, type = "response")

#threshold
pred_classes <- ifelse(pred_probs >= 0.5, 1, 0)
pred_classes <- as.factor(pred_classes)

test_data$protest_attend <- as.factor(test_data$protest_attend)
pred_classes <- factor(pred_classes, levels = levels(test_data$protest_attend))

confusionMatrix(pred_classes, test_data$protest_attend, positive = "1")
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction   0   1
##          0 201  98
##          1 121 180
##
##                Accuracy : 0.635
##                  95% CI : (0.5951, 0.6736)
##     No Information Rate : 0.5367
##     P-Value [Acc > NIR] : 6.934e-07
##
##                   Kappa : 0.2702
##
##  Mcnemar's Test P-Value : 0.1371
##
##             Sensitivity : 0.6475
##             Specificity : 0.6242
##          Pos Pred Value : 0.5980
##          Neg Pred Value : 0.6722
##              Prevalence : 0.4633
##          Detection Rate : 0.3000
##    Detection Prevalence : 0.5017
##       Balanced Accuracy : 0.6359
##
##        'Positive' Class : 1
##
```

This model's accuracy is about 65.7%, which is similar to our initial random forest model. This suggests that our core predictors alone are again not strong indicators of protest participation. To improve the model, we used logistic regression to identify the variables with the highest coefficients.

```r
glm_model <- glm(protest_attend ~ ., data = protestData1, family = binomial)
coefs <- summary(glm_model)$coefficients

top20 <- coefs[order(abs(coefs[ , "Estimate"]), decreasing = TRUE)[1:20], ]

top20
```

```
##                                     Estimate   Std. Error       z value
## (Intercept)                    -4.610951e+06 1.000474e+11 -4.608766e-05
## mesreal                        -1.080191e+03 1.749025e+07 -6.175959e-05
## superven                       -2.392769e+02 3.036068e+06 -7.881145e-05
## trustinarmedforces             -2.334649e+02 3.056325e+06 -7.638745e-05
## sexo                            2.209966e+02 2.843184e+06  7.772857e-05
## lifesatisfaction               -1.827077e+02 2.432498e+06 -7.511117e-05
## economysatisfaction             1.813334e+02 2.948053e+06  6.150956e-05
## countrynext12monthseconsituation -1.696032e+02 2.077754e+06 -8.162814e-05
## trustinpeople                   1.451891e+02 1.362308e+06  1.065759e-04
## totperd                         1.290833e+02 2.068532e+06  6.240337e-05
## countryeconomicsituation        1.234006e+02 1.637556e+06  7.535654e-05
## supervvi                        1.106805e+02 2.670815e+06  4.144071e-05
## totrevi                        -9.458007e+01 1.541722e+06 -6.134705e-05
## tamciud                         9.046346e+01 1.578691e+06  5.730284e-05
## trustinpolice                   8.654138e+01 1.054214e+06  8.209090e-05
```

```
## familynext12monthseconsituation   7.014508e+01 1.620573e+06  4.328411e-05
## democracypreference               5.777624e+01 6.110267e+05  9.455600e-05
## democracysatisfaction            -5.776172e+01 8.535313e+05 -6.767381e-05
## codif                             5.460210e+01 1.053822e+06  5.181339e-05
## diareal                          -4.857811e+01 7.503357e+05 -6.474184e-05
##                                   Pr(>|z|)
## (Intercept)                      0.9999632
## mesreal                          0.9999507
## superven                         0.9999371
## trustinarmedforces               0.9999391
## sexo                             0.9999380
## lifesatisfaction                 0.9999401
## economysatisfaction              0.9999509
## countrynext12monthseconsituation 0.9999349
## trustinpeople                    0.9999150
## totperd                          0.9999502
## countryeconomicsituation         0.9999399
## supervvi                         0.9999669
## totrevi                          0.9999511
## tamciud                          0.9999543
## trustinpolice                    0.9999345
## familynext12monthseconsituation  0.9999655
## democracypreference              0.9999246
## democracysatisfaction            0.9999460
## codif                            0.9999587
## diareal                          0.9999483
```

```
relevant_vars <- c(
  "trustinarmedforces", "sexo", "lifesatisfaction", "economysatisfaction",
  "countrynext12monthseconsituation", "trustinpeople",
  "countryeconomicsituation", "trustinpolice"
)

model_df <- protestData1[, c(relevant_vars, "protest_attend")]

#imputing missing values
imputed1 <- mice(model_df, method = "pmm", seed = 123)
```

```
##
##  iter imp variable
##   1   1  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   1   2  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   1   3  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   1   4  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   1   5  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   2   1  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   2   2  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   2   3  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   2   4  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   2   5  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   3   1  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   3   2  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   3   3  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   3   4  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
```

```
##   3   5  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   4   1  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   4   2  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   4   3  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   4   4  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   4   5  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   5   1  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   5   2  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   5   3  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   5   4  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
##   5   5  trustinarmedforces  lifesatisfaction  economysatisfaction  countrynext12monthseconsituation
```

```r
completed_df1 <- complete(imputed1)
completed_df1$protest_attend <- as.factor(protestData1$protest_attend)

set.seed(123)
sample_index1 <- sample(nrow(completed_df1), size = 0.5 * nrow(completed_df1))
train_data1 <- completed_df1[sample_index1, ]
test_data1 <- completed_df1[-sample_index1, ]

selected_formula <- as.formula(paste("protest_attend ~", paste(relevant_vars, collapse = " + ")))

logit_model <- glm(selected_formula, data = train_data1, family = "binomial")

pred_probs <- predict(logit_model, newdata = test_data1, type = "response")
pred_classes <- factor(ifelse(pred_probs >= 0.5, 1, 0), levels = c(0, 1))

test_data$protest_attend <- factor(test_data$protest_attend, levels = c(0, 1))
confusionMatrix(pred_classes, test_data$protest_attend, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 167  76
##          1 155 202
##
##                Accuracy : 0.615
##                  95% CI : (0.5747, 0.6541)
##     No Information Rate : 0.5367
##     P-Value [Acc > NIR] : 6.460e-05
##
##                   Kappa : 0.2406
##
##  Mcnemar's Test P-Value : 2.866e-07
##
##             Sensitivity : 0.7266
##             Specificity : 0.5186
##          Pos Pred Value : 0.5658
##          Neg Pred Value : 0.6872
##              Prevalence : 0.4633
##          Detection Rate : 0.3367
##    Detection Prevalence : 0.5950
##       Balanced Accuracy : 0.6226
```

```
##
##        'Positive' Class : 1
##
```

After pulling the top 20 and removing internal or non-informative survey items, we found that trust in the armed forces, gender, life satisfaction, satisfaction with the economy, outlook on the country's economy, trust in people, and trust in police were among the most important predictors. We then imputed missing values using predictive mean matching, split the data into training and testing sets, and fit another logistic regression model. The overall accuracy for this model dropped slightly to 61%. Why did the accuracy decrease? We hypothesize that the lower accuracy may reflect logistic regression's limitations in capturing complex, nonlinear patterns in the data. Moreover, logistic regression assumes that predictors contribute independently to the outcome. In our case, this may oversimplify the complex relationships between attitudes, demographics, and protest behavior, so we may be missing some important interactions between variables.

Still, logistic regression helps understand which individual predictors are most strongly associated with protest behavior. It was a significant finding that economic satisfaction and life satisfaction came up again as strong predictors because random forest found economic score (which is a a combined measure of these and other economy-related variables), so we can deduce that a person's perception of their economic situation and the country help determine their likelihood of protest. This suggests that a person's perception of both their personal financial situation and the country's broader economic outlook plays an important role in their likelihood of protesting. This insight aligns with the context of Chile's 2019 protests, which were largely driven by public frustration over economic inequality and rising living costs.

As for our research question of whether individual attitudes toward democracy, the economy, trust, and freedom can predict protest participation, our findings suggest that perceptions of the economy are particularly strong predictors of Chileans' likelihood to protest, based on results from both models.