dataAnalysis.R

mollyofferwestort

2025-10-24

Install and load required packages

```
#install.packages(c("caret", "randomForest", "sandwich", "lmtest", "stargazer"))
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(kableExtra)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:kableExtra':
##
##     group_rows
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(estimatr)
library(modelsummary)
```

```
## `modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing
##   backend. Learn more at: https://vincentarelbundock.github.io/tinytable/
##
## Revert to `kableExtra` for one session:
##
##   options(modelsummary_factory_default = 'kableExtra')
##   options(modelsummary_factory_latex = 'kableExtra')
##   options(modelsummary_factory_html = 'kableExtra')
##
## Silence this message forever:
```

```
##
##   config_modelsummary(startup_message = FALSE)
```

```
library(grf)
```

Set up

```
# Setup ####
```

Functions

```
# for formatting in tables
f1 <- function(x) format(round(x, 3))
formula <- as.formula("post_test ~ treatment_frame")

# cleaning names for regression tables
rename_treatment_frame <- function(old_names) {
  new_names <- gsub("treatment_frame", "", old_names)
  new_names <- gsub("(Intercept)", "Control mean", new_names)
  setNames(new_names, old_names)
}

options(modelsummary_factory_latex = 'kableExtra')
```

Import data

```
data <- read.csv("./../data/data_with_additional_vars.csv", as.is = TRUE)

treatments <- c("No framing", "Negative science", "Religious", "Equity", "Efficiency", "Secular")
data$treatment_frame <- factor(data$treatment_frame, levels = treatments)
data_rep <- data[data$party == 1,]
data_dem <- data[data$party == -1,]
data_ind <- data[data$party == 0,]

set.seed(42)
num_folds <- 5
covariates_pre <- c("gastax", "carbtax", "treaty", "regcarb")

outcome_var <- "post_test"
covariates <- c("age", "party_id", "employment_status", "race_white", "income_level",
                "relationship", "college", "sex_id", "prosociality", "gastax",
                "carbtax", "treaty", "regcarb", "ideology", "scientific_confidence",
                "reward_consequence", "religiosity", "rel_freq", "economic_reasoning")
treatment_vars <- paste0("treatment_", 1:5)
control_var <- "pre_test"
```

Distribution of subjects across treatment conditions

```
# Treatment distribution ####
result_df <- count(data, treatment_value, treatment_frame) |>
  mutate(proportion = n / sum(n)) |>
  rename(count = n)
result_df
```

```
##   treatment_value  treatment_frame count proportion
```

```
## 1               0       No framing   396  0.1689420
## 2               1 Negative science   395  0.1685154
## 3               2        Religious   358  0.1527304
## 4               3           Equity   402  0.1715017
## 5               4       Efficiency   411  0.1753413
## 6               5          Secular   382  0.1629693
```

Pre-post test means tables

```
# Pre/post-test Means ####
means_table <- data  |>
  group_by(treatment_value, treatment_frame) |>
  summarise(pre_test_est = mean(pre_test, na.rm = TRUE),
            pre_test_se = sd(pre_test, na.rm = TRUE)/sqrt(n()),
            post_test_est = mean(post_test, na.rm = TRUE),
            post_test_se = sd(post_test, na.rm = TRUE)/sqrt(n()))
```

```
## `summarise()` has grouped output by 'treatment_value'. You can override using
## the `.groups` argument.
```

```
means_estimates <- means_table |>
  select(treatment_frame, pre_test_est, post_test_est) |>
  mutate(`Pre test` = paste0("\\num{", round(pre_test_est, 3), "}"),
         `Post test` = paste0("\\num{", round(post_test_est, 3), "}")) |>
  select(-pre_test_est, -post_test_est)
```

```
## Adding missing grouping variables: `treatment_value`
```

```
se_estimates <- means_table |>
  select(treatment_frame, pre_test_se, post_test_se) |>
  mutate(`Pre test` = paste0("(\\num{", round(pre_test_se, 3), "})"),
         `Post test` = paste0("(\\num{", round(post_test_se, 3), "})")) |>
  select(-pre_test_se, -post_test_se)
```

```
## Adding missing grouping variables: `treatment_value`
```

```
means_latex <- bind_rows(means_estimates, se_estimates, .id = "id") |>
  arrange(treatment_frame) |>
  mutate(Treatment = case_when(id == 2 ~ "",
                               TRUE ~ treatment_frame)) |>
  ungroup() |>
  select(Treatment, `Pre test`, `Post test`)

# Convert to LaTeX using kableExtra
kbl(means_latex,
    # Title and table caption for LaTeX
    caption= "Mean response estimates by treatment group. \\label{tab:means}", # table title and label
    # Add nicely grouped lines under column names
    midrule = "\\cmidrule(lr){2-3}",
    align = c("l", "c", "c"), # Align the columns (one for treatment name, two for response columns)
    booktabs = TRUE, # Use booktabs styling
    format = "latex",
    escape = FALSE, # Don't escape LaTeX special characters
    linesep = c("", "\\addlinespace")) |> # Add space between rows
  kable_styling(full_width = FALSE, # Don't use full width
                latex_options = c("HOLD_position")) |> # LaTeX float options
```

```
  footnote(general = # Add a note at the bottom of the table
             paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{"
                    nrow(data),
                    "}. Columns represent  averages of policy index questions, pre- and post- delivery o
           escape = FALSE,
           threeparttable = TRUE,
           general_title = "") |>
  save_kable("../tables/means_table.tex")
```

By political party

```
# * Means by Party ####
party_means_table <- data  |>
  group_by(treatment_value, treatment_frame, party) |>
  mutate(party = case_when(party == 1 ~ "Republican",
                           party == -1 ~ "Democrat",
                           party == 0 ~ "Independent")) |>
  summarise(pre_test_est = mean(pre_test, na.rm = TRUE),
            pre_test_se = sd(pre_test, na.rm = TRUE)/sqrt(n()),
            post_test_est = mean(post_test, na.rm = TRUE),
            post_test_se = sd(post_test, na.rm = TRUE)/sqrt(n()))
```

```
## `summarise()` has grouped output by 'treatment_value', 'treatment_frame'. You
## can override using the `.groups` argument.
```

```
party_means_estimates <- party_means_table |>
  select(treatment_frame, pre_test_est, post_test_est, party) |>
  mutate(`Pre test` = paste0("\\num{", round(pre_test_est, 3), "}"),
         `Post test` = paste0("\\num{", round(post_test_est, 3), "}")) |>
  select(-pre_test_est, -post_test_est)
```

```
## Adding missing grouping variables: `treatment_value`
```

```
party_se_estimates <- party_means_table |>
  select(treatment_frame, pre_test_se, post_test_se, party) |>
  mutate(`Pre test` = paste0("(\\num{", round(pre_test_se, 3), "})"),
         `Post test` = paste0("(\\num{", round(post_test_se, 3), "})")) |>
  select(-pre_test_se, -post_test_se)
```

```
## Adding missing grouping variables: `treatment_value`
```

```
party_means_latex <- bind_rows(party_means_estimates, party_se_estimates,
                               .id = "id") |>
  arrange(treatment_frame) |>
  mutate(Treatment = case_when(id == 2 ~ "",
                               TRUE ~ treatment_frame)) |>
  ungroup() |>
  pivot_wider(names_from = party, values_from = c(`Pre test`, `Post test`)) |>
  select(Treatment, `Pre test_Democrat`, `Post test_Democrat`,
         `Pre test_Independent`, `Post test_Independent`,
         `Pre test_Republican`, `Post test_Republican`)

# Convert to LaTeX using kableExtra
kbl(party_means_latex,
    col.names = c("Treatment", rep(c("Pre-test", "Post-test"), 3)),  # Update column names
```

```
    caption= "Mean response estimates by party identification. \\label{tab:party_means}",
    align = c("l", rep("c", 6)),  # Align the columns (one for treatment, six for responses)
    # Add nicely grouped lines under column names
    midrule = "\\cmidrule(lr){2-3} \\cmidrule(lr){4-5} \\cmidrule(lr){6-7}",
    format = "latex",
    booktabs = TRUE,  # Use booktabs styling
    escape = FALSE, # Don't escape LaTeX special characters
    linesep = c("", "\\addlinespace"), # Add space between rows
    table.envir = "table*") |> # Special environment so table can span 2 columns
  # Add in the first level of column headers: Democrats, Independents, Republicans
  add_header_above(c(" " = 1, "Democrat" = 2, "Independent" = 2, "Republican" = 2),
                   line = FALSE) |>
  kable_styling(full_width = FALSE,  # Don't use full width
                latex_options = c("hold_position")) |>  # LaTeX float options
  # Add a note at the bottom of the table
  footnote(general = paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \
                            nrow(data),
                            "}. Columns represent averages of policy index questions, pre- and post- de
           escape = FALSE,
           threeparttable = TRUE,
           general_title = "") |>
  save_kable("../tables/party_means_table.tex")
```

Estimates of treatment effects

```
# Estimation ####
```

Difference in means

```
# * D-I-M ####
model0 <- lm_robust(formula, data = data, se = "HC3")
```

Lin estimates with all covariates

```
# * Lin all covariates ####
model1 <- lm_lin(formula,
                 covariates =
                   formula(paste0("~ ",
                                  paste(covariates, collapse = " + "))),
                 data = data, se = "HC3")
```

Lin estimates with pre-test covariates only

```
# * Lin pre-test only ####
model2 <- lm_lin(formula,
                 covariates =
                   formula(paste0("~ ",
                                  paste(covariates_pre, collapse = " + "))),
                 data = data, se = "HC3")
```

Table

```
# ** ATE estimates table generation ####
# Drop statistical significance for control means
model0$p.value[1] <- model1$p.value[1] <- model2$p.value[1] <- NA
model0$statistic[1] <- model1$statistic[1] <- model2$statistic[1] <- NA


modelsummary(list(`Difference-in-means` = model0,
                  `Adjusted (pre-test only)` = model2,
                  `Adjusted (all)` = model1),
             align = "lccc",
             output = 'latex',
             coef_omit = "_c", # exclude all of the controls and interactions
             coef_rename = rename_treatment_frame,
             gof_omit = ".*",
             stars = TRUE ,
             estimate="{estimate}{stars}",
             escape = FALSE,
             title= 'Treatment effect estimates and response.\\label{tab:treatment_effects}',
             table.envir = "table*") |>
  kable_styling(latex_options = c('HOLD_position')) |>
  footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{",
                  nrow(data),
                  "}. Estimates are average treatment effects on the post-test policy index measure as
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '') |>
  save_kable("../tables/means_linear_regressions.tex")
```

```
## Warning: To compile a LaTeX document with this table, the following commands must be placed in the d
##
## \usepackage{tabularray}
## \usepackage{float}
## \usepackage{graphicx}
## \usepackage{codehigh}
## \usepackage[normalem]{ulem}
## \UseTblrLibrary{booktabs}
## \UseTblrLibrary{siunitx}
## \newcommand{\tinytableTabularrayUnderline}[1]{\underline{#1}}
## \newcommand{\tinytableTabularrayStrikeout}[1]{\sout{#1}}
## \NewTableCommand{\tinytableDefineColor}[3]{\definecolor{#1}{#2}{#3}}
##
## To disable `siunitx` and prevent `modelsummary` from wrapping numeric entries in `\num{}`, call:
##
## options("modelsummary_format_numeric_latex" = "plain")
##  This warning appears once per session.
```

Best fixed and personalized treatments

```
# * Best fixed and personalized ####
# simple difference in means table
data$best_treatment_factor <- as.factor(case_when(
  data$treatment_value == data$best_fixed_arm ~ 1,
  data$treatment_value != data$best_fixed_arm & data$treatment_value != 0 ~ 2,
```

```r
  TRUE ~ 0
))

# Create best_personalized_treatment_indicator
data$best_personalized_factor <- as.factor(case_when(
  data$treatment_value == data$best_personalized_arm ~ 1,
  data$treatment_value != data$best_personalized_arm & data$treatment_value != 0 ~ 2,
  TRUE ~ 0
))

data$best_fixed_personalized_factor <- as.factor(case_when(
  data$treatment_value == data$best_fixed_arm & data$treatment_value == data$best_personalized_arm ~ 3,
  data$treatment_value == data$best_fixed_arm ~ 1,
  data$treatment_value == data$best_personalized_arm ~ 2,
  data$treatment_value != data$best_personalized_arm & data$treatment_value != data$best_fixed_arm & da
  TRUE ~ 0
))
# Create models
model_fixed <- lm_robust(post_test ~ best_treatment_factor, data = data, se_type = "HC3")
model_personalized <- lm_robust(post_test ~ best_personalized_factor, data = data, se_type = "HC3")

# relevel for comparisons
data <- data |>
  mutate(best_treatment_factor_releveled = relevel(best_treatment_factor, ref = "2"))

model_fixed2 <- lm_robust(post_test ~ best_treatment_factor_releveled, data = data, se_type = "HC3")

# grf estimates
out_list <- list()
scores_list <- list()
for(k in sort(unique(data$fold))){
  data_k <- data[data$fold == k,]
  # we did selection on the data not in the kth fold
  # so we get OOB estimates on data in the kth fold
  # Causal forest estimates

  gg_fixed <- multi_arm_causal_forest(Y = data_k$post_test,
                                      W = as.factor(data_k$best_treatment_factor),
                                      X = as.matrix(data_k[, covariates]))
  gg_fixed2 <- multi_arm_causal_forest(Y = data_k$post_test,
                                       W = as.factor(data_k$best_treatment_factor_releveled),
                                       X = as.matrix(data_k[, covariates]))
  gg_personalized <- multi_arm_causal_forest(Y = data_k$post_test,
                                             W = as.factor(data_k$best_personalized_factor),
                                             X = as.matrix(data_k[, covariates]))
  scores_fixed <- get_scores(gg_fixed)
  scores_fixed2 <- get_scores(gg_fixed2)
  scores_personalized <- get_scores(gg_personalized)

  scores_on <- data.frame(best_fixed_est_grf = scores_fixed[,,1][,1],
                          best_personalized_est_grf = scores_personalized[,,1][,1],
                          best_difference_est_grf = scores_fixed[,,1][,1] - scores_personalized[,,1][,1
                          not_best_fixed_est_grf = scores_fixed2[,,1][,2])
```

```
  # add to the data
  out_list[[k+1]] <- scores_on
}
```

Table

```
# ** Best estimates table generation ####

# save for reference for estimates
best_means_table <- do.call(rbind.data.frame, out_list) |>
  summarize(across(.cols = everything(),
                   .fns = list(
                     mean = ~mean(.x, na.rm = TRUE),
                     se = ~ sd(.x, na.rm = TRUE)/sqrt(n()),
                     stat = ~ mean(.x, na.rm = TRUE)/(sd(.x, na.rm = TRUE)/sqrt(n())),
                     pval = ~ 2 * (1 - pnorm(abs(mean(.x, na.rm = TRUE)/(sd(.x, na.rm = TRUE)/sqrt(n())
                   )))

best_means_table_formatted <- do.call(rbind.data.frame, out_list) |>
  summarize(across(.cols = everything(),
                   .fns = list(
                     mean_formatted = ~ paste0("\\num{", round(mean(.x, na.rm = TRUE), 3), "}",
                                               case_when(
                                                 abs(mean(.x, na.rm = TRUE)/(sd(.x, na.rm = TRUE)/sqrt(n
                                                 abs(mean(.x, na.rm = TRUE)/(sd(.x, na.rm = TRUE)/sqrt(n
                                                 abs(mean(.x, na.rm = TRUE)/(sd(.x, na.rm = TRUE)/sqrt(n
                                                 abs(mean(.x, na.rm = TRUE)/(sd(.x, na.rm = TRUE)/sqrt(n
                                                 TRUE ~ "***")),
                     se_formatted = ~ paste0("(\\num{", round(sd(.x, na.rm = TRUE)/sqrt(n()), 3), "})")
                   ))) |>
  pivot_longer(cols = everything(),
               names_to = c("Treatment", "statistic"),
               names_pattern = "([a-z]+)_([a-z]+)") |>
  mutate(Treatment = c("Best fixed", "",
                       "Best personalized", "",
                       "Fixed - personalized", "",
                       "Fixed - sub-optimal", ""))

best_means_dim <- data.frame("Difference in means" = c(
  # best fixed
  paste0("\\num{", round(coef(summary(model_fixed))[2,1], 3), "}",
         case_when(abs(coef(summary(model_fixed))[2,3]) < 1.65 ~ "",
                   abs(coef(summary(model_fixed))[2,3]) < 1.96 ~ "+",
                   abs(coef(summary(model_fixed))[2,3]) < 2.58 ~ "*",
                   abs(coef(summary(model_fixed))[2,3]) < 3.29 ~ "**",
                   TRUE ~ "***")),
  paste0("(\\num{", round(coef(summary(model_fixed))[2,2], 3), "})"),
  # best personalized
  paste0("\\num{", round(coef(summary(model_personalized))[2,1], 3), "}",
         case_when(abs(coef(summary(model_personalized))[2,3]) < 1.65 ~ "",
                   abs(coef(summary(model_personalized))[2,3]) < 1.96 ~ "+",
                   abs(coef(summary(model_personalized))[2,3]) < 2.58 ~ "*",
                   abs(coef(summary(model_personalized))[2,3]) < 3.29 ~ "**",
```

```
                    TRUE ~ "***")),
  paste0("(\\num{", round(coef(summary(model_personalized))[2,2], 3), "})"),
  # best fixed - best personalized
  paste0("\\num{", round(coef(summary(model_fixed))[2,1] - coef(summary(model_personalized))[2,1], 3),
         case_when(abs((coef(summary(model_fixed))[2,1] - coef(summary(model_personalized))[2,1])/sqrt(
                   abs((coef(summary(model_fixed))[2,1] - coef(summary(model_personalized))[2,1])/sqrt(
                   abs((coef(summary(model_fixed))[2,1] - coef(summary(model_personalized))[2,1])/sqrt(
                   abs((coef(summary(model_fixed))[2,1] - coef(summary(model_personalized))[2,1])/sqrt(
                   TRUE ~ "***")),
  paste0("(\\num{", round( sqrt((coef(summary(model_fixed))[2,2]^2 + coef(summary(model_personalized))[
  # best fixed - suboptimal
  paste0("\\num{", round(coef(summary(model_fixed2))[2,1], 3), "}",
         case_when(abs(coef(summary(model_fixed2))[2,3]) < 1.65 ~ "",
                   abs(coef(summary(model_fixed2))[2,3]) < 1.96 ~ "+",
                   abs(coef(summary(model_fixed2))[2,3]) < 2.58 ~ "*",
                   abs(coef(summary(model_fixed2))[2,3]) < 3.29 ~ "**",
                   TRUE ~ "***")),
  paste0("(\\num{", round(coef(summary(model_fixed2))[2,2], 3), "})")
),
check.names = FALSE)

best_means_latex <- bind_cols(best_means_table_formatted[,"Treatment"],
                              best_means_dim,
                              best_means_table_formatted[, "value"],
)

# Create LaTeX table
kbl(best_means_latex,
    col.names = c("", "Difference-in-means", "Fold-wise causal forests"),  # Update column names
    caption = "Treatment effect estimates for best fixed and best personalized arms. \\label{tab:best_f
    align = c("lcc"),
    midrule = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3}",
    linesep = c("", "\\addlinespace"),
    booktabs = TRUE,
    format = "latex",
    escape = FALSE,
    table.envir = "table*") |> # Special environment so table can span 2 columns
  row_spec(5, extra_latex_after = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3}") |>
  kable_styling(latex_options = c("hold_position"),
                full_width = FALSE) |>
  footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{",
                  nrow(data),
                  "}. Estimates are average treatment effects as compared to the control on the post-tes
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '') |>
  save_kable("../tables/best_framing_table.tex")

# ** GRF fold estimates table generation ####
fold_average_predictions_table <- do.call(rbind.data.frame, out_list) |>
  summarise(across(.cols = everything(),
                   .fns = list(
                     mean = ~ paste0("\\num{", round(mean(.x, na.rm = TRUE), 3), "}",
```

```
                                    case_when(
                                      abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 3.291 ~ "***",
                                      abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 2.576 ~ "**",
                                      abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 1.960 ~ "*",
                                      abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 1.645 ~ ".",
                                      TRUE ~ "***")),
                    se = ~ paste0("(\\num{", round(sd(.x)/sqrt(n()), 3), "})")
                  ), .names = "{.col}_{.fn}")) |>
  pivot_longer(cols = everything(),
               names_to = c("Treatment", "statistic"),
               names_sep = "_est_grf_") |>
  mutate(Treatment = c("Best fixed", "",
                       "Best personalized", "",
                       "Fixed - personalized", "",
                       "Fixed - sub-optimal", ""))

fold_predictions_table <- do.call(rbind.data.frame, out_list) |>
  mutate(fold = data$fold) |>
  group_by(fold) |>
  summarise(across(.cols = everything(),
                   .fns = list(
                     mean = ~ paste0("\\num{", round(mean(.x, na.rm = TRUE), 3), "}",
                                     case_when(
                                       abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 3.291 ~ "***",
                                       abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 2.576 ~ "**",
                                       abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 1.960 ~ "*",
                                       abs(mean(.x, na.rm = TRUE) / (sd(.x, na.rm = TRUE) / sqrt(n()))) >= 1.645 ~ ".",
                                       TRUE ~ "***")),
                     se = ~ paste0("(\\num{", round(sd(.x)/sqrt(n()), 3), "})")
                   ), .names = "{.col}_{.fn}")) |>
  pivot_longer(cols = -fold,
               names_to = c("Treatment", "statistic"),
               names_sep = "_est_grf_") |>
  pivot_wider(names_from= fold, values_from = value, names_glue = "Fold {fold}") |>
  mutate(Treatment = c("Best fixed", "",
                       "Best personalized", "",
                       "Fixed - personalized", "",
                       "Fixed - sub-optimal", ""))

# Convert to LaTeX using kableExtra
kbl(bind_cols(fold_predictions_table[, -2], `Fold average` = fold_average_predictions_table$value),
    caption= "Mean average treatment effect estimates by fold. \\label{tab:fold_estimates}",
    align = c("l", rep("c", 6)),  # Align the columns (one for treatment, 6 for folds + average)
    # Add nicely grouped lines under column names
    midrule = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3} \\cmidrule(lr){4-4} \\cmidrule(lr){5-5} \\cmidrul
    format = "latex",
    booktabs = TRUE,  # Use booktabs styling
    escape = FALSE, # Don't escape LaTeX special characters
    linesep = c("", "\\addlinespace"), # Add space between rows
    table.envir = "table*") |> # Special environment so table can span 2 columns
  kable_styling(full_width = FALSE,  # Don't use full width
                latex_options = c("hold_position")) |>  # LaTeX float options
  # Add a note at the bottom of the table
```

```
  footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{",
                  nrow(data),
                  "}, split in to five equal folds. Estimates are average treatment effects as compared
           escape = FALSE,
           threeparttable = TRUE,
           general_title = "") |>
  save_kable("../tables/fold_estimates_table.tex")
```

Summary statistics and balance tables

```
# Summary statistics ####

var_list <- c(
  pre_test = "Policy support (pre-test)",
  gastax = "Gas tax support (pre-test)",
  carbtax = "Carbon tax support (pre-test)",
  treaty = "Treaty support (pre-test)",
  post_test = "Policy support (post-test)",
  age = "Age",
  college = "College degree",
  party_dem = "Democrat",
  party_rep = "Republican",
  party_ind = "Independent",
  sex_id = "Male",
  race_white = "White",
  relationship = "Relationship",
  prosociality = "Prosociality",
  religiosity = "Religiosity",
  rel_freq = "Religious frequency",
  economic_reasoning = "Economic reasoning"
)

# Create derived party dummies
data$party_dem <- as.numeric(data$party == -1)
data$party_rep <- as.numeric(data$party == 1)
data$party_ind <- as.numeric(data$party == 0)

# Subset relevant variables
summary_data <- data[names(var_list)]

# Compute summary statistics
summary_stats <- data.frame(
  Variable = unname(var_list),
  Mean = sapply(summary_data, function(x) mean(x, na.rm = TRUE)),
  SD = sapply(summary_data, function(x) sd(x, na.rm = TRUE)),
  Min = sapply(summary_data, function(x) min(x, na.rm = TRUE)),
  Max = sapply(summary_data, function(x) max(x, na.rm = TRUE))
)

rownames(summary_stats) <- NULL

summary_stats$Mean <- sprintf("%.3f", summary_stats$Mean)
summary_stats$SD <- sprintf("%.3f", summary_stats$SD)
```

```r
summary_stats$Min <- sprintf("%.3f", summary_stats$Min)
summary_stats$Max <- sprintf("%.3f", summary_stats$Max)

#Create LaTeX table
kbl(summary_stats,
    caption = "Summary Statistics \\label{tab:summary_stats}",
    col.names = c("", "Mean", "SD", "Min", "Max"),
    align = c("l", "c", "c", "c", "c"),
    booktabs = TRUE,
    format = "latex",
    escape = TRUE,
    table.envir = "table*") |> # Special environment so table can span 2 columns) |>
  kable_styling(latex_options = c("hold_position"),
                full_width = FALSE) |>
  save_kable("../tables/summary_stats_table.tex")
```

Balance Table

```r
balance_vars <- var_list[names(var_list) != "post_test"]

#Initialize empty dataframe
balance_table <- data.frame(
  Variable = character(),
  stringsAsFactors = FALSE
)

# Loop through balance variables
for (i in seq_along(balance_vars)) {
  var <- names(balance_vars[i])
  label <- balance_vars[[i]]

  means_by_treatment <- data |>
    group_by(treatment_frame) |>
    summarise(
      mean = mean(.data[[var]], na.rm = TRUE),
      se = sd(.data[[var]], na.rm = TRUE) / sqrt(sum(!is.na(.data[[var]])))
    )

  # Format rows: one for mean, one for SE
  row_mean <- c(label)
  row_se <- c("")

  for (t in treatments) {
    m <- means_by_treatment |> filter(treatment_frame == t)
    row_mean <- c(row_mean, sprintf("%.3f", m$mean))
    row_se <- c(row_se, sprintf("(%.3f)", m$se))
  }

  balance_table <- rbind(balance_table, row_mean, row_se)
}

# Create and save LaTeX table
kbl(balance_table,
    caption = "Balance Table: Means by Treatment Condition \\label{tab:balance}",
```

```
    col.names = c("Variable", treatments),
    booktabs = TRUE,
    format = "latex",
    align = c('l', rep('c', length(treatments))),
    escape = TRUE,
    linesep = "",
    table.envir = "table*") |>
  kable_styling(latex_options = c("hold_position", "scale_down"),
                full_width = FALSE) |>
  footnote(general = "Note: Standard errors are reported in parentheses below estimates.",
           escape = FALSE,
           threeparttable = TRUE,
           general_title = "") |>
  save_kable("../tables/balance_table.tex")
```

By political party

```
# Party ####
party_means_table <- data  |>
  group_by(treatment_value, treatment_frame, party) |>
  mutate(party = case_when(party == 1 ~ "Republican",
                           party == -1 ~ "Democrat",
                           party == 0 ~ "Independent")) |>
  summarise(pre_test_est = mean(pre_test, na.rm = TRUE),
            pre_test_se = sd(pre_test, na.rm = TRUE)/sqrt(n()),
            post_test_est = mean(post_test, na.rm = TRUE),
            post_test_se = sd(post_test, na.rm = TRUE)/sqrt(n()))
```

```
## `summarise()` has grouped output by 'treatment_value', 'treatment_frame'. You
## can override using the `.groups` argument.
```

```
party_means_estimates <- party_means_table |>
  select(treatment_frame, pre_test_est, post_test_est, party) |>
  mutate(`Pre test` = paste0("\\num{", round(pre_test_est, 3), "}"),
         `Post test` = paste0("\\num{", round(post_test_est, 3), "}")) |>
  select(-pre_test_est, -post_test_est)
```

```
## Adding missing grouping variables: `treatment_value`
```

```
party_se_estimates <- party_means_table |>
  select(treatment_frame, pre_test_se, post_test_se, party) |>
  mutate(`Pre test` = paste0("(\\num{", round(pre_test_se, 3), "})"),
         `Post test` = paste0("(\\num{", round(post_test_se, 3), "})")) |>
  select(-pre_test_se, -post_test_se)
```

```
## Adding missing grouping variables: `treatment_value`
```

```
party_means_latex <- bind_rows(party_means_estimates, party_se_estimates,
                               .id = "id") |>
  arrange(treatment_frame) |>
  mutate(Treatment = case_when(id == 2 ~ "",
                               TRUE ~ treatment_frame)) |>
  ungroup() |>
  pivot_wider(names_from = party, values_from = c(`Pre test`, `Post test`)) |>
  select(Treatment, `Pre test_Democrat`, `Post test_Democrat`,
         `Pre test_Independent`, `Post test_Independent`,
```

```
        `Pre test_Republican`, `Post test_Republican`)

# Convert to LaTeX using kableExtra
kbl(party_means_latex,
    col.names = c("Treatment", rep(c("Pre-test", "Post-test"), 3)),  # Update column names
    caption= "Mean response estimates by party identification. \\label{tab:party_means}",
    align = c("l", rep("c", 6)),  # Align the columns (one for treatment, six for responses)
    # Add nicely grouped lines under column names
    midrule = "\\cmidrule(lr){2-3} \\cmidrule(lr){4-5} \\cmidrule(lr){6-7}",
    format = "latex",
    booktabs = TRUE,  # Use booktabs styling
    escape = FALSE, # Don't escape LaTeX special characters
    linesep = c("", "\\addlinespace"), # Add space between rows
    table.envir = "table*") |> # Special environment so table can span 2 columns
  # Add in the first level of column headers: Democrats, Independents, Republicans
  add_header_above(c(" " = 1, "Democrat" = 2, "Independent" = 2, "Republican" = 2),
                   line = FALSE) |>
  kable_styling(full_width = FALSE,  # Don't use full width
                latex_options = c("hold_position")) |>  # LaTeX float options
  # Add a note at the bottom of the table
  footnote(general = paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \
                            nrow(data),
                            "}. Columns represent averages of policy index questions, pre- and post- de
           escape = FALSE,
           threeparttable = TRUE,
           general_title = "") |>
  save_kable("../tables/party_means_table.tex")
```

Party: Difference in Means

```
# * Party: D-I-M ####
#Separate models for Rep, Dem, and Ind
model0_rep <- lm_robust(formula, data = data_rep, se = "HC3")
model0_dem <- lm_robust(formula, data = data_dem, se = "HC3")
model0_ind <- lm_robust(formula, data = data_ind, se = "HC3")

# Drop statistical significance for control means
model0_dem$p.value[1] <- model0_ind$p.value[1] <- model0_rep$p.value[1] <- NA
model0_dem$statistic[1] <- model0_ind$statistic[1] <- model0_rep$statistic[1] <- NA

modelsummary(list(Democrat = model0_dem,
                  Independent = model0_ind,
                  Republican = model0_rep),
             output = 'latex',
             midrule = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3} \\cmidrule(lr){4-4}",
             coef_rename = rename_treatment_frame,
             gof_omit = ".*",
             stars = TRUE ,
             estimate="{estimate}{stars}",
             gof_map = list(list('raw' = 'nobs',
                                 'clean' = 'n',
                                 'fmt' = f1)),
             escape = FALSE,
```

```
            title= 'Treatment effect estimates and response by party identification.\\label{tab:party_
            table.envir = "table*") |>
  kable_styling(latex_options = c('HOLD_position')) |>
  footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{",
                  nrow(data),
                  "}. Estimates are average treatment effects on the post-test policy index measure as
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '') |>
  save_kable("../tables/party_dim_table.tex")
```

Party: Lin estimates with all covariates

```
# * Party: Lin all covariates ####

#Separate models for Rep, Dem, and Ind
model1_rep <- lm_lin(formula,
                     covariates =
                       formula(paste0("~ ",
                                      paste(covariates, collapse = " + "))),
                     data = data_rep, se = "HC3")
model1_dem <- lm_lin(formula,
                     covariates =
                       formula(paste0("~ ",
                                      paste(covariates, collapse = " + "))),
                     data = data_dem, se = "HC3")
model1_ind <- lm_lin(formula,
                     covariates =
                       formula(paste0("~ ",
                                      paste(covariates, collapse = " + "))),
                     data = data_ind, se = "HC3")

# Drop statistical significance for control means
model1_dem$p.value[1] <- model1_ind$p.value[1] <- model1_rep$p.value[1] <- NA
model1_dem$statistic[1] <- model1_ind$statistic[1] <- model1_rep$statistic[1] <- NA

modelsummary(list(Democrat = model1_dem,
                  Independent = model1_ind,
                  Republican = model1_rep),
             output = 'latex',
             coef_omit = "_c", # exclude all of the controls and interactions
             coef_rename = rename_treatment_frame,
             midrule = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3} \\cmidrule(lr){4-4}",
             stars = TRUE ,
             estimate="{estimate}{stars}",
             gof_omit = ".*",
             align = "lccc",
             gof_map = list(list('raw' = 'nobs',
                                 'clean' = 'n',
                                 'fmt' = f1)),
             escape = FALSE,
             title= 'Treatment effect estimates and response by party identification, controlling for p
             table.envir = "table*") |>
```

```
  kable_styling(latex_options = c('HOLD_position')) |>
  footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{",
                  nrow(data),
                  "}. Estimates are average treatment effects on the post-test policy index measure as
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '') |>
  save_kable("../tables/party_lin_all_table.tex")


# * Party: Lin pre-test only ####
```

Party: Lin estimates with pre-test covariates only

```
#Separate models for Rep, Dem, and Ind
model2_rep <- lm_lin(formula,
                     covariates =
                       formula(paste0("~ ",
                                      paste(covariates_pre, collapse = " + "))),
                     data = data_rep, se = "HC3")
model2_dem <- lm_lin(formula,
                     covariates =
                       formula(paste0("~ ",
                                      paste(covariates_pre, collapse = " + "))),
                     data = data_dem, se = "HC3")
model2_ind <- lm_lin(formula,
                     covariates =
                       formula(paste0("~ ",
                                      paste(covariates_pre, collapse = " + "))),
                     data = data_ind, se = "HC3")

# Drop statistical significance for control means
model2_dem$p.value[1] <- model2_ind$p.value[1] <- model2_rep$p.value[1] <- NA
model2_dem$statistic[1] <- model2_ind$statistic[1] <- model2_rep$statistic[1] <- NA

modelsummary(list(Democrat = model2_dem,
                  Independent = model2_ind,
                  Republican = model2_rep),
             output = 'latex',
             coef_omit = "_c", # exclude all of the controls and interactions
             coef_rename = rename_treatment_frame,
             midrule = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3} \\cmidrule(lr){4-4}",
             stars = TRUE ,
             estimate="{estimate}{stars}",
             gof_omit = ".*",
             align = "lccc",
             gof_map = list(list('raw' = 'nobs',
                                 'clean' = 'n',
                                 'fmt' = f1)),
             escape = FALSE,
             title= 'Treatment effect estimates and response by party identification, controlling for p
             table.envir = "table*") |>
  kable_styling(latex_options = c('HOLD_position')) |>
```

```
  footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is all respondents, $n = $ \\\\num{",
                  nrow(data),
                  "}. Estimates are average treatment effects on the post-test policy index measure as
           escape = FALSE,
           threeparttable = TRUE,
           general_title = '') |>
  save_kable("../tables/party_lin_table.tex")
```

Party: Best fixed and personalized treatments

```
# * Party: Best fixed and personalized ####
# simple difference in means table

# Run best arm analysis separately for each subgroup (Democrats, Republicans, Independents)

# Helper functions to format coefficients and SEs with stars
format_coef <- function(model, i) {
  est <- coef(summary(model))[i, 1]
  se <- coef(summary(model))[i, 2]
  stat <- est / se
  stars <- case_when(
    abs(stat) < 1.65 ~ "",
    abs(stat) < 1.96 ~ "+",
    abs(stat) < 2.58 ~ "*",
    abs(stat) < 3.29 ~ "**",
    TRUE ~ "***"
  )
  paste0("\\num{", round(est, 3), "}", stars)
}

format_se <- function(model, i) {
  se <- coef(summary(model))[i, 2]
  paste0("(\\num{", round(se, 3), "})")
}

format_diff <- function(model1, model2) {
  est1 <- coef(summary(model1))[2, 1]
  est2 <- coef(summary(model2))[2, 1]
  diff <- est1 - est2
  se1 <- coef(summary(model1))[2, 2]
  se2 <- coef(summary(model2))[2, 2]
  stat <- diff / sqrt(se1^2 + se2^2)
  stars <- case_when(
    abs(stat) < 1.65 ~ "",
    abs(stat) < 1.96 ~ "+",
    abs(stat) < 2.58 ~ "*",
    abs(stat) < 3.29 ~ "**",
    TRUE ~ "***"
  )
  paste0("\\num{", round(diff, 3), "}", stars)
}

format_diff_se <- function(model1, model2) {
```

```r
  se1 <- coef(summary(model1))[2, 2]
  se2 <- coef(summary(model2))[2, 2]
  se_combined <- sqrt(se1^2 + se2^2)
  paste0("(\\num{", round(se_combined, 3), "})")
}

run_best_arm_analysis <- function(df, group_label, table_path = "../tables") {
  treat0 <- treatments[treatments != "No framing"]
  df$best_fixed_arm <- NA
  df$best_personalized_arm <- NA

  # for each fold i = 1 to k do
  for (fold in unique(df$fold)) {
    # Define:
    # • Training set Dtrain = D \ Di,
    # • Test set Dtest = Di
    df_train <- df[df$fold != fold, ]
    df_test <- df[df$fold == fold, ]

    # Stage 1: Best Fixed treatment selection
    # (1a) Fit a linear model on the outcome using Dtrain with predictors:
    # • Treatment indicators, pre-test response, and their interaction
    lm_fixed <- lm_lin(formula,
                       covariates = formula(paste0("~ ", paste(covariates_pre, collapse = " + "))),
                       data = df_train, se = "HC3")

    # (1b) Predict the response for Dtest under each treatment
    outv <- sapply(treat0, function(t) {
      df_test$treatment_frame <- t
      mean(predict(lm_fixed, df_test))
    })

    # (1c) Save treatment arm with highest predicted outcome
    df[df$fold == fold, "best_fixed_arm"] <- which.max(outv)

    # Stage 2: Best Personalized treatment selection
    # (2a) Train separate RF models for each treatment condition on Dtrain
    outmat <- matrix(NA, nrow = nrow(df_test), ncol = length(treat0))
    for (i in seq_along(treat0)) {
      df_train_t <- df_train[df_train$treatment_frame == treat0[i], ]
      grf <- regression_forest(Y = df_train_t$post_test,
                               X = as.matrix(df_train_t[, covariates]))
      outmat[, i] <- predict(grf, newdata = as.matrix(df_test[, covariates]))$predictions
    }
    # (2c) Save treatment with highest predicted outcome for each obs
    df[df$fold == fold, "best_personalized_arm"] <- apply(outmat, 1, which.max)
  }

  # Stage 3: Estimation of Average Treatment Effects
  df <- df |>
    mutate(
      best_treatment_factor = as.factor(case_when(
        treatment_value == best_fixed_arm ~ 1,
```

```
      treatment_value != best_fixed_arm & treatment_value != 0 ~ 2,
      TRUE ~ 0
    )),
    best_personalized_factor = as.factor(case_when(
      treatment_value == best_personalized_arm ~ 1,
      treatment_value != best_personalized_arm & treatment_value != 0 ~ 2,
      TRUE ~ 0
    )),
    best_fixed_personalized_factor = as.factor(case_when(
      treatment_value == best_fixed_arm & treatment_value == best_personalized_arm ~ 3,
      treatment_value == best_fixed_arm ~ 1,
      treatment_value == best_personalized_arm ~ 2,
      treatment_value != 0 ~ 4,
      TRUE ~ 0
    )),
    best_treatment_factor_releveled = relevel(best_treatment_factor, ref = "2")
  )

# Create models
model_fixed <- lm_robust(post_test ~ best_treatment_factor, data = df, se_type = "HC3")
model_personalized <- lm_robust(post_test ~ best_personalized_factor, data = df, se_type = "HC3")
model_fixed2 <- lm_robust(post_test ~ best_treatment_factor_releveled, data = df, se_type = "HC3")

# Causal forest estimates by fold
out_list <- list()
for (k in sort(unique(df$fold))) {
  data_k <- df[df$fold == k, ]
  gg_fixed <- multi_arm_causal_forest(Y = data_k$post_test,
                                      W = as.factor(data_k$best_treatment_factor),
                                      X = as.matrix(data_k[, covariates]))
  gg_fixed2 <- multi_arm_causal_forest(Y = data_k$post_test,
                                       W = as.factor(data_k$best_treatment_factor_releveled),
                                       X = as.matrix(data_k[, covariates]))
  gg_personalized <- multi_arm_causal_forest(Y = data_k$post_test,
                                             W = as.factor(data_k$best_personalized_factor),
                                             X = as.matrix(data_k[, covariates]))
  scores_fixed <- get_scores(gg_fixed)
  scores_fixed2 <- get_scores(gg_fixed2)
  scores_personalized <- get_scores(gg_personalized)

  scores_on <- data.frame(
    best_fixed_est_grf = scores_fixed[,,1][,1],
    best_personalized_est_grf = scores_personalized[,,1][,1],
    best_difference_est_grf = scores_fixed[,,1][,1] - scores_personalized[,,1][,1],
    not_best_fixed_est_grf = scores_fixed2[,,1][,2]
  )

  out_list[[k + 1]] <- scores_on
}

best_means_table_formatted <- do.call(rbind, out_list) |>
  summarize(across(.cols = everything(),
                   .fns = list(
```

```
                      mean_formatted = ~ paste0("\\num{", round(mean(.x, na.rm = TRUE), 3), "}",
                                                case_when(
                                                  abs(mean(.x)/(sd(.x, na.rm = TRUE)/sqrt(n()))) < 1.65
                                                  abs(mean(.x)/(sd(.x, na.rm = TRUE)/sqrt(n()))) < 1.96
                                                  abs(mean(.x)/(sd(.x, na.rm = TRUE)/sqrt(n()))) < 2.58
                                                  abs(mean(.x)/(sd(.x, na.rm = TRUE)/sqrt(n()))) < 3.29
                                                  TRUE ~ "***")),
                      se_formatted = ~ paste0("(\\num{", round(sd(.x)/sqrt(length(.x)), 3), "})")
                    ))) |>
    pivot_longer(cols = everything(),
                 names_to = c("Treatment", "statistic"),
                 names_pattern = "([a-z_]+)_([a-z]+)") |>
    mutate(Treatment = c("Best fixed", "",
                         "Best personalized", "",
                         "Fixed - personalized", "",
                         "Fixed - sub-optimal", ""))

  best_means_dim <- data.frame("Difference in means" = c(
    format_coef(model_fixed, 2),
    format_se(model_fixed, 2),
    format_coef(model_personalized, 2),
    format_se(model_personalized, 2),
    format_diff(model_fixed, model_personalized),
    format_diff_se(model_fixed, model_personalized),
    format_coef(model_fixed2, 2),
    format_se(model_fixed2, 2)
  ), check.names = FALSE)

  best_means_latex <- bind_cols(best_means_table_formatted[,"Treatment"],
                                best_means_dim,
                                best_means_table_formatted[, "value"])

  # Save LaTeX table with full footnote
  save_kable(
    kbl(best_means_latex,
        col.names = c("", "Difference-in-means", "Fold-wise causal forests"),
        caption = paste0("Treatment effect estimates for best fixed and best personalized arms, ", group
        align = c("lcc"),
        midrule = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3}",
        linesep = c("", "\\addlinespace"),
        booktabs = TRUE,
        format = "latex",
        escape = FALSE,
        table.envir = "table*") |>
      row_spec(5, extra_latex_after = "\\cmidrule(lr){2-2} \\cmidrule(lr){3-3}") |>
      kable_styling(latex_options = c("hold_position"), full_width = FALSE) |>
      footnote(paste0("\\\\footnotesize \\\\textit{Note:} The sample is ", group_label, ", $n = $ \\\\nu
                      "}. Estimates are average treatment effects as compared to the control on the post
               escape = FALSE, threeparttable = TRUE, general_title = ''),
    file = file.path(table_path, paste0("best_framing_table_", tolower(group_label), ".tex"))
  )
}
```

```r
# Run for each subgroup
datasets <- list(Democrats = data_dem, Republicans = data_rep, Independents = data_ind)

purrr::walk2(datasets, names(datasets), run_best_arm_analysis)



# Robustness ####
# Drop failed attention checks
data_check <- data |>
  filter(attention_check_1pass == 1 & attention_check_2pass == 1)

party_means_table <- data_check  |>
  group_by(treatment_value, treatment_frame, party) |>
  mutate(party = case_when(party == 1 ~ "Republican",
                           party == -1 ~ "Democrat",
                           party == 0 ~ "Independent")) |>
  summarise(pre_test_est = mean(pre_test, na.rm = TRUE),
            pre_test_se = sd(pre_test, na.rm = TRUE)/sqrt(n()),
            post_test_est = mean(post_test, na.rm = TRUE),
            post_test_se = sd(post_test, na.rm = TRUE)/sqrt(n()))
```

```
## `summarise()` has grouped output by 'treatment_value', 'treatment_frame'. You
## can override using the `.groups` argument.
```

```r
party_means_estimates <- party_means_table |>
  select(treatment_frame, pre_test_est, post_test_est, party) |>
  mutate(`Pre test` = paste0("\\num{", round(pre_test_est, 3), "}"),
         `Post test` = paste0("\\num{", round(post_test_est, 3), "}")) |>
  select(-pre_test_est, -post_test_est)
```

```
## Adding missing grouping variables: `treatment_value`
```

```r
party_se_estimates <- party_means_table |>
  select(treatment_frame, pre_test_se, post_test_se, party) |>
  mutate(`Pre test` = paste0("(\\num{", round(pre_test_se, 3), "})"),
         `Post test` = paste0("(\\num{", round(post_test_se, 3), "})")) |>
  select(-pre_test_se, -post_test_se)
```

```
## Adding missing grouping variables: `treatment_value`
```

```r
party_means_latex <- bind_rows(party_means_estimates, party_se_estimates,
                               .id = "id") |>
  arrange(treatment_frame) |>
  mutate(Treatment = case_when(id == 2 ~ "",
                               TRUE ~ treatment_frame)) |>
  ungroup() |>
  pivot_wider(names_from = party, values_from = c(`Pre test`, `Post test`)) |>
  select(Treatment, `Pre test_Democrat`, `Post test_Democrat`,
         `Pre test_Independent`, `Post test_Independent`,
         `Pre test_Republican`, `Post test_Republican`)


# failed attention check by party
data_failed <- data |>
  group_by(party) |>
```

```r
  summarize(fail_rate = paste0("\\num{", round(100 * mean(attention_check_1pass == 0 | attention_check_1
  ) |>
  pivot_wider(names_from = party, values_from = fail_rate)

failed_row <- tibble(
  Treatment = "Failed attention check (percent)",
  `Pre test_Democrat` = data_failed$`-1`,
  `Post test_Democrat` = "",
  `Pre test_Independent` = data_failed$`0`,
  `Post test_Independent` = "",
  `Pre test_Republican` = data_failed$`1`,
  `Post test_Republican` = ""
)

# Append to party_means_latex
party_means_latex <- bind_rows(
  party_means_latex,
  failed_row
)


# Convert to LaTeX using kableExtra
kbl(party_means_latex,
    col.names = c("Treatment", rep(c("Pre-test", "Post-test"), 3)),  # Update column names
    caption= "Mean response estimates by party identification. \\label{tab:party_means_passed}",
    align = c("l", rep("c", 6)),  # Align the columns (one for treatment, six for responses)
    # Add nicely grouped lines under column names
    midrule = "\\cmidrule(lr){2-3} \\cmidrule(lr){4-5} \\cmidrule(lr){6-7}",
    format = "latex",
    booktabs = TRUE,  # Use booktabs styling
    escape = FALSE, # Don't escape LaTeX special characters
    linesep = c("", "\\addlinespace"), # Add space between rows
    table.envir = "table*") |> # Special environment so table can span 2 columns
  # Add in the first level of column headers: Democrats, Independents, Republicans
  add_header_above(c(" " = 1, "Democrat" = 2, "Independent" = 2, "Republican" = 2),
                   line = FALSE) |>
  kable_styling(full_width = FALSE,  # Don't use full width
                latex_options = c("hold_position")) |>  # LaTeX float options
  # Add a note at the bottom of the table
  footnote(general = paste0("\\\\footnotesize \\\\textit{Note:} The sample is respondents that passed b
                            nrow(data_check),
                            "}. Columns represent averages of policy index questions, pre- and post- de
           escape = FALSE,
           threeparttable = TRUE,
           general_title = "") |>
  save_kable("../tables/party_means_table_passed.tex")
```