STATS 209
Introduction to Causal Inference

Reassessing Causal Relationships: Investigating the Impact of Deliberative Minipublics on Public Opinion with a Focus on Participant Political Affiliation

Andri Vidarsson, Eden Luvishis, Veronica Yakubovich

Instructor: Dominik Rothenhäusler
December 12, 2023

1 Introduction

As we approach the upcoming presidential election cycle and gear up for campaign season in the next few months, a variety of campaign strategies will soon be in full swing. From negative campaigning to media outreach and celebrity endorsements, these approaches raise intriguing questions about how public opinion takes shape. In this project, we aim to explore individuals' receptiveness to altering their views on a politically sensitive topic when presented with the collective perspective of a small assembly of ordinary citizens who've thoroughly examined the subject. We build on the paper *"Can Deliberative Minipublics Influence Public Opinion? Theory and Experimental Evidence"*. Deliberative mini publics are, as defined in the paper, "small groups of citizens who deliberate together about a policy issue and convey their conclusions to decision makers" (Ingham & Levin, 2018). Leveraging the data presented in the referenced paper, our objective is to assess the presence of a causal relationship between individuals being exposed to the endorsement of a deliberative minipublic and their stance on the specific issue under consideration. Notably, our analysis will be conducted separately for each major political party. We will be using data from the published study but re-analyzing it with causal inference techniques. More specifically, we will be using standard Neymanian inference methods and covariate adjustment using Lin's estimator to determine if there exists a causal relationship between a deliberative mini public cue and voter opinion. Our findings suggest that the impact of being presented with a deliberative minipublic's conclusions is most noticeable among individuals who don't align with any political party, that is, independents.

2 Experimental Overview

In their 2018 study, the authors implemented a completely randomized experiment (CRE) involving participants from the state of California. The participants were assigned to one of three treatments or a control group and were subsequently surveyed regarding their stance on a proposed tax amendment to the State Legislature. This amendment aimed to reduce the required supermajority threshold for passing new taxes from 67% to 55%. The three treatments involved exposure to different informational cues, delivered through video and text, with the intent of influencing participants' support for the amendment. Treatment arms included:

- A control group receiving no cue (Arm I);
- A group receiving the deliberative cue (Arm II), conveyed through a video and text describing deliberations among participants in a deliberative poll, where most approved of the lower majority requirement;
- A group receiving the party cue (Arm III), framed as follows: "Democrats tend to approve of this proposal, while Republicans tend to disapprove of it";
- And a treatment arm receiving both cues (Arm IV).

Following exposure to these treatments, participants were solicited to express their support for the amendment using a 5-point Likert scale, ranging from Strongly Disapprove to Strongly Approve.

While the study design is robust and yields intriguing findings, the ensuing analysis in the paper is notably limited. The reported difference-in-means and regression outcomes are confined in scope. Therefore, we propose a comprehensive reanalysis of the experiment's data, employing advanced causal techniques. This reanalysis will account for pre-treatment covariates, treatment exposures, and appropriately address the ordinal nature of the data. Our focus will be on two of the four treatments: Arm III, where participants received the party cue, and Arm IV, where participants received both the party and deliberative cues, denoted as $z_i \in \{0, 1\}$. We opt to exclude Arms I and II from our analysis, positing that, even without explicit exposure to the 'party cue' during the experiment, most participants may possess awareness of the Democratic and Republican positions on tax law issues (Pew Research Center, 2012). Moreover, our analysis emphasizes the treatment effects within each political party since individuals' perspectives on government procedures and taxes are closely tied to their political affiliations (Pew Research Center, 2015). Consequently, we investigate whether a deliberative minipublic has the potential to influence the viewpoints of those already aligned with a political party—an aspect we consider a significant signal in itself. Essentially, this approach mirrors post-stratification, as we intend to examine the treatment effect within each party (Democrat, Republican, and Independent) separately.

3 Data

The experiment's initial survey had 1,750 participants and collected information on a participant's level of agreement with the new tax proposal and information on covariates, including party affiliation, gender, age, income, education, as well as if the participant was white or hispanic.

Arm I	Arm II	Arm III	Arm IV	Total
No cues	Deliberative cue	Party cue	Deliberative and Party cue	-
427	408	425	415	1675

Table 1: Participants in each Arm of the experiment

After excluding Arms I and II and dropping participants who did not indicate a level of agreement or had data for income, our analysis will be conducted on the remaining 840 participants in Arms III and IV (Table 1). To facilitate analysis, in the EDA stage, Z and Y variables were introduced, with Z denoting 0 for participants exposed solely to the party cue (Arm III) and 1 for those exposed to both cues (Arm IV). The dependent variable Y indicated the level of approval with the proposal on a discrete scale from -2 to 2, where -2 and 2 corresponded to strong disapproval and strong approval respectively, and 0 signified neutrality. In addition, the columns corresponding to education and party were labeled accordingly and the income variable was binned to facilitate visualization.

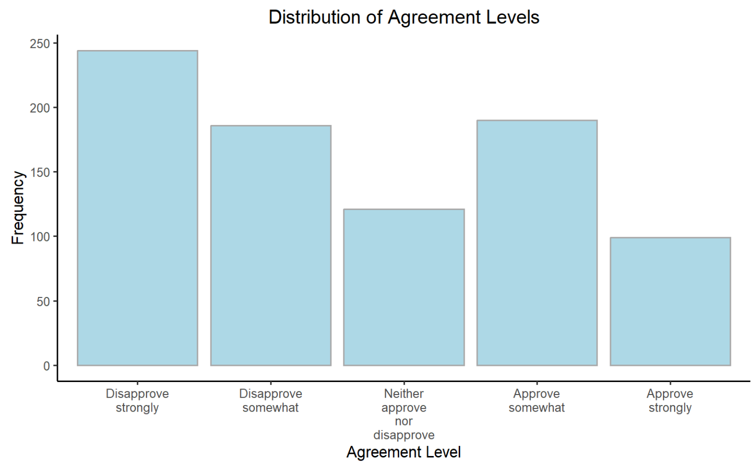


Figure 1: Distribution of outcomes

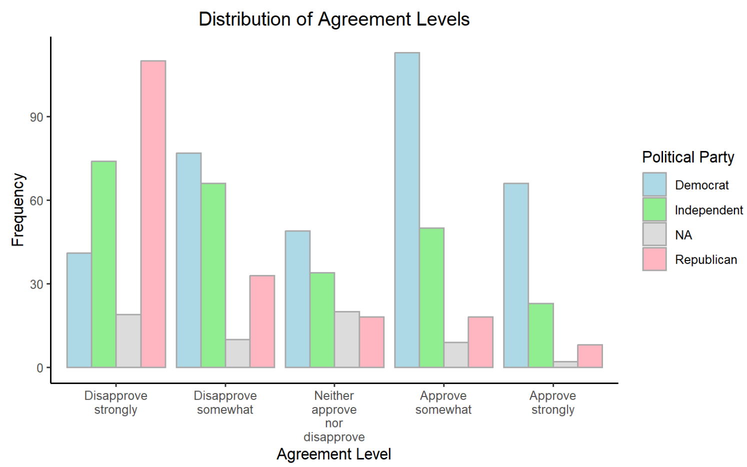


Figure 2: Distribution of outcomes by political party

Initially, looking at the distribution of agreement levels, without differentiating by the treatment or political party (Fig. 1), it appears that most participants disapprove either strongly or somewhat with the proposal, with the lowest in the strongly approve category. However, when breaking down by party (Fig. 2), we can see differences in opinion split by party line, particularly when looking at Democrats and Republicans in the somewhat approve and disapprove categories. We will explore this difference further and see if there is a causally significant relationship.

4 Methods

In the original study, the authors analyzed the data using a difference in means estimator and a linear model studying the interaction between different treatments (Ingham & Levin, 2018). While they looked at all treatment arms across all data, we propose to specifically look at strata of the experiment based on participant political party.

Notably, the outcomes of the experiment (support of the tax amendment to the state legislature) have a unique structure that will affect our analysis. The 5-point Likert Scale rating introduces potentially limited information about participant opinion with or without treatment. Currently, there exist a number of approaches to deal with ordinal outcomes in causal experiments, including using a discrete scale, binarizing the data, as well as simulating a quasi-continuous distribution and building nonlinear models to approximate it (Boes, 2013). For the rest of our methodology, however, we choose to linearly map the outcomes to a range $\in [-2, 2]$ in order to compare our methods to the findings in the original paper.

4.1 FRT

We begin our analysis with a Fisher Randomization Test (FRT) with a choice of a difference-in-means test statistic for all of the data–without stratification into political parties. Our treatment effect and corresponding p-value are defined as follows (Ding, 2023):

$$\hat{\tau} = \hat{Y}(1) - \hat{Y}(0)$$

, $$\text{where } \hat{Y}(1) = n_1^{-1} \textstyle\sum_{i=1}^{n} Z_i Y_i, \text{ and } \hat{Y}(0) = n_0^{-1} \textstyle\sum_{i=1}^{n} (1 - Z_i) Y_i$$

$$\hat{p}_{FRT} = R^{-1} \sum_{r=1}^{R} I\{T(z^{\tau}, Y) \geq T(Z, Y)\}$$

The FRT allows us to examine whether the treatment (in the form of the deliberative cue) has a statistically significant effect, or whether the outcomes are only a result of randomization.

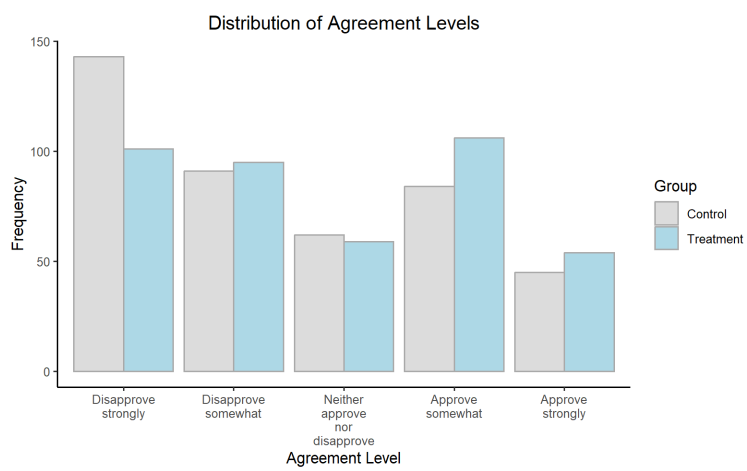


Figure 3: Distribution of outcomes for control and treatment groups

We can see that there is, even visually (Fig. 3), a significant change in distribution of outcomes for treatment and control groups, so we expect a low p-value for the FRT. Therefore, we will conduct other analyses to confirm our findings.

4.2 Neymanian Inference

In addition to our FRT analysis, we wanted to determine how confident we can be in our difference-in-means estimate, which we will call $\hat{\tau}$. To do this, we calculated Neymanian estimates of variance and then computed 95% confidence intervals (CIs).

Instead of only looking at CIs for the whole dataset, we aim to examine the effects of the deliberative cue within each political party separately (Fig. 4).

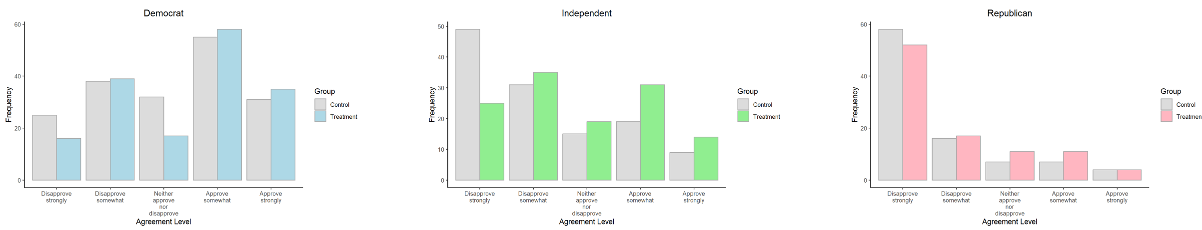


Figure 4: Distribution of outcomes by party.

We hypothesize that the treatment may affect members of different parties in different ways. These confidence intervals will help us see if we have enough evidence to conclude whether the average treatment effect (ATE) is significantly different from zero. Under Neyman's null, we use the same estimator of $\hat{\tau}$ as in the FRT and the following conservative estimator of variance of $\hat{\tau}$ and confidence interval:

$$\hat{V}ar[\hat{\tau}] = n_1^{-1}\hat{S}^2(1) + n_0^{-1}\hat{S}^2(0)$$

$$, \quad \text{where } \hat{S}^2(1) = (n_1 - 1)^{-1}\textstyle\sum_{i:Z_i=1}(Y_i - n_1^{-1}\sum_{j:Z_j} Y_j)^2$$

$$\text{and } \hat{S}^2(0) = (n_0 - 1)^{-1}\textstyle\sum_{i:Z_i=0}(Y_i - n_0^{-1}\sum_{j:Z_j} Y_j)^2$$

$$95\%\ CI : \hat{\tau} \pm 1.96\sqrt{\hat{V}ar[\hat{\tau}]}$$

4.3 Covariate Adjustment with Lin's Estimator

The rationale for employing Lin's estimator, denoted as $\hat{\tau}_{pred}$, lies in its effectiveness in addressing covariate imbalances. It achieves this by minimizing the impact of covariate imbalances by residualizing potential outcomes (Ding, 2023). In particular, we adjust for the covariates of income, education, age, and race, as studies suggest these have a correlation with opinions on taxes and government (Pew Research Center, 2019). We use a generalized version of Lin's estimator with a random forest model as a predictor since the relationship between the covariates and the outcome does not appear to be linear (Figure 5).

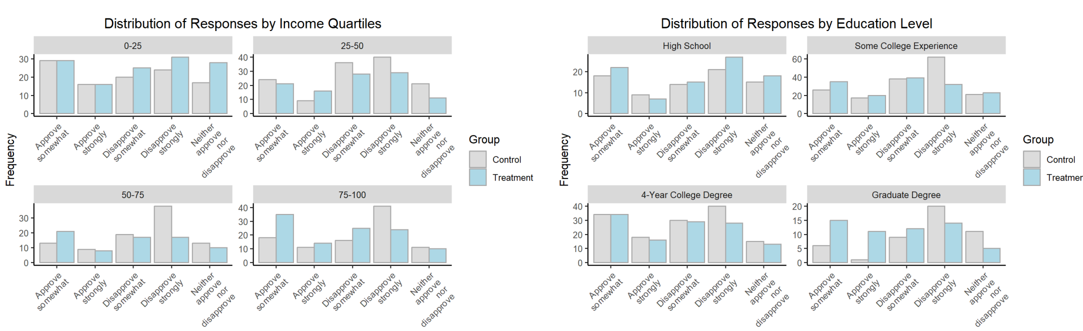


Figure 5: Distribution of outcomes by income and education level.

Below are formulas for Lin's treatment effect estimator with an underlying random forest model where the variance is estimated using the bootstrap method (Ding, 2023):

$$\hat{\tau}_{pred} = n^{-1}\sum_{i=1}^{n} Z_iY_i + (1-Z_i)\tilde{\mu}_1(x_i) - n^{-1}\sum_{i=1}^{n}(1-Z_i)Y_i + Z_i\tilde{\mu}_0(x_i)$$

where $\tilde{\mu}_1(x) = \hat{\mu}_1(x) + n_1^{-1}\sum_{i:Z_i=1} Y_i - \hat{\mu}_1(x_i)$ and $\tilde{\mu}_0(x) = \hat{\mu}_0(x) + n_0^{-1}\sum_{i:Z_i=0} Y_i - \hat{\mu}_0(x_i)$
where $\hat{\mu}_1(x)$ and $\hat{\mu}_1(x)$ are predicted using random forest
$\hat{V}ar_{pred}$ = Approximated with bootstrap, i.e. sample variance of estimates of $\hat{\tau}_{pred}$ from repeated resampling.

$$95\%CI : \hat{\tau}_{pred} \pm 1.96\sqrt{\hat{V}ar_{pred}}$$

5 Results

5.1 FRT

A Fisher's randomization test conducted on the entire dataset (all political parties included) reveals a positive treatment effect associated with receiving the deliberative cue, supported by a significantly low p-value. However, it's crucial to acknowledge the potential limitations of this method in the context of our experiment. Given that the potential outcomes can only assume five discrete values, it becomes improbable for the p-value not to be very low, unless there is no noticeable difference in means between the treatment and control for the observed data. In essence, even a subtle increase in support may not be adequately captured by the randomized treatment assignments in a Fisher's randomization test, resulting in an exceptionally low p-value.

$\hat{\tau}$	p-value
0.2696496	0.0041

Table 2: Results from FRT on entire dataset

5.2 Neymanian inference

Neymanian confidence intervals were computed, first on the entire dataset, and then for each group separately.

$\hat{\tau}$	CI
0.2696496	(0.0715136, 0.4677856)

Table 3: Neymanian confidence interval for entire dataset

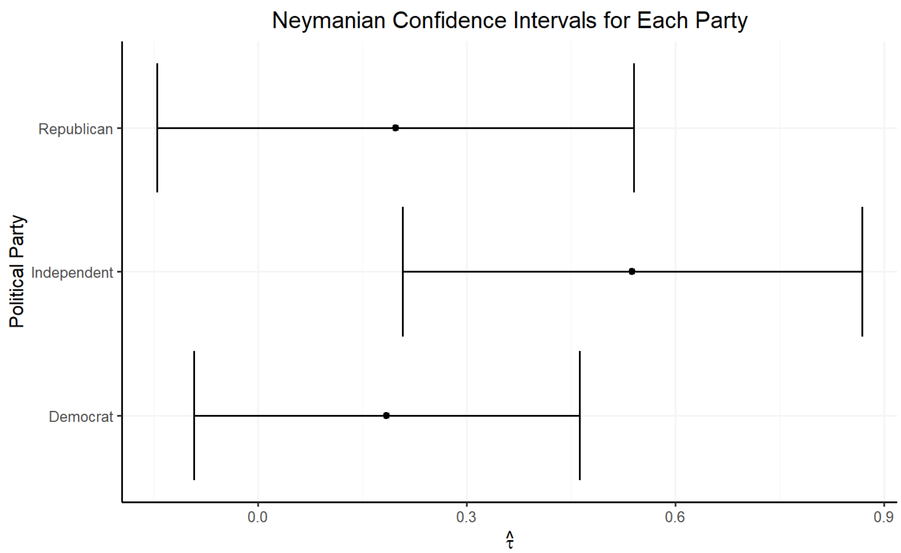


Figure 6: Neymanian confidence interval per political party

These results match those in the paper. When the data is observed as a whole, there seems to be a statistically significant treatment effect. However, when analyzed for each group separately, there is only a significant treatment effect for independents.

5.3 Covariate adjustment with generalized Lin's Estimator

$\hat{\tau}$	CI
0.1876604	(-0.0498380 0.4251588)

Table 4: Normal approximated confidence interval for Lin's estimator with boostrapped variance

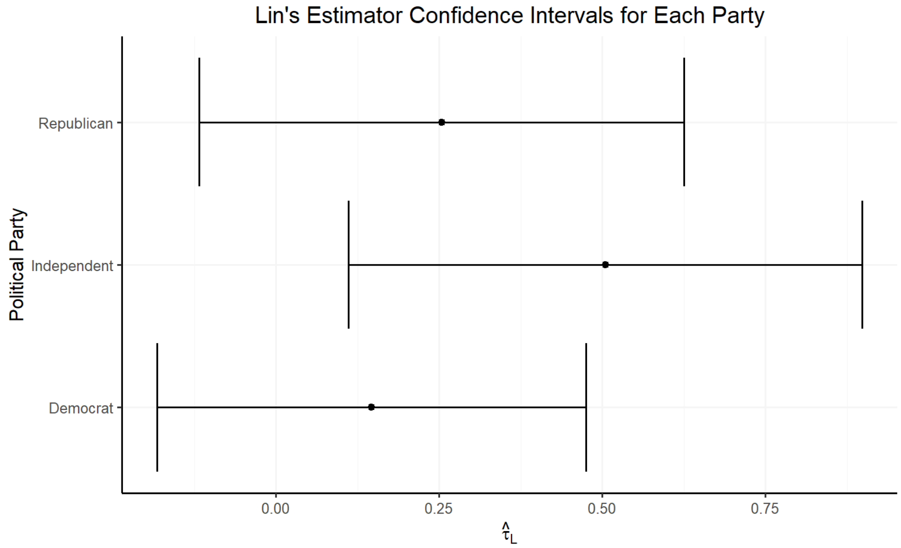


Figure 7: Confidence intervals for Lin's estimator with boostrapped variane, per political party

With covariate adjustment, in contrast to the results from the standard Neymanian inference there is not a significant treatment effect when the data is analyzed as a whole. When the Lin's model is run on each group separately, we get the same results as for standard Neymanian approach, and there is only a significant treatment effect for independents.

6 Conclusion

Our analysis suggests that there is only a significant effect from receiving a deliberative cue for those individuals that are not registered to a political party. However, it is crucial to approach these results with caution, considering the inherent sensitivity of our survey topic, which aligns with the partisan divide. Given the polarized nature of political opinions, it becomes imperative to acknowledge the limitations of generalizing our findings across the entire population. To enhance the robustness and generalizability of future research, we suggest a study design where a deliberate-minipublic weighs in on a diverse range of topics and presents its conclusion on each of those topics to the treatment arm. This approach can provide a more comprehensive understanding of the dynamics at play, allowing for a nuanced exploration of treatment effects across various subjects.

In summary, our analysis, validated by Fisher's Randomization Test, Neymanian estimates, and Lin's estimator, identifies a clear treatment effect among independents. To fortify our results, exploring diverse modeling approaches for ordinal and discrete outcomes is essential. While our current linear mapping suffices, more sophisticated techniques can capture sensitivity to treatment with greater nuance.

In essence, our study provides a foundation for deeper investigations into treatment effects, result generalizability, and refined modeling, offering insights into how deliberative cues influence opinions and impact public policy and democratic processes.

References

Boes, S. (2013). Nonparametric analysis of treatment effects in ordered response models. *Empirical Economics*, *44*, 81–109. Retrieved from https://doi.org/10.1007/s00181-010-0354-y doi: 10.1007/s00181-010-0354-y

Ding, P. (2023). *A first course in causal inference.*

Ingham, S., & Levin, I. (2018). Can deliberative minipublics influence public opinion? theory and experimental evidence. *Political Research Quarterly*, *71*(3), 654–667. Retrieved 2023-11-12, from http://www.jstor.org/stable/45106689

Pew Research Center. (2012). *What the public knows about the political parties.* Retrieved from https://www.pewresearch.org/politics/2012/04/11/what-the-public-knows-about-the-political-parties/

Pew Research Center. (2015). *General opinions about the federal government.* Retrieved from https://www.pewresearch.org/politics/2015/11/23/2-general-opinions-about-the-federal-government/

Pew Research Center. (2019). *Domestic policy: Taxes, environment, health care.* Retrieved from https://www.pewresearch.org/politics/2019/12/17/7-domestic-policy-taxes-environment-health-care/

Appendix - Code

We have included the following three files which were used to process the raw data, conduct EDA, and conduct causal inference:

- dataProceesing.Rmd - returns a processed and cleaned data set to be used in the analysis as described in the Data section
- EDA.Rmd - generates graphs in the Data, Methods, and Results sections
- Inference.Rmd - produces the results, graphs and analysis for the Results section

dataProcessing.R

Andri Vidarsson, Eden Luvishis, Veronica Yakubovich

```
loadData = function(){
  load("dlexpdata.rdata")
  dlexpdata$response <- NA
  dlexpdata$response[dlexpdata$agree == "Approve strongly"] <- 2
  dlexpdata$response[dlexpdata$agree == "Approve somewhat"] <- 1
  dlexpdata$response[dlexpdata$agree == "Neither approve nor disapprove"] <- 0
  dlexpdata$response[dlexpdata$agree == "Disapprove somewhat"] <- -1
  dlexpdata$response[dlexpdata$agree == "Disapprove strongly"] <- -2

  dlexp <- filter(dlexpdata, !is.na(response), !is.na(income))

  incQuants = quantile(dlexp$income)
  dlexp$incomeQuant <- NULL
  dlexp$incomeQuant[dlexp$income <= incQuants[2]] <- "0-25"
  dlexp$incomeQuant[dlexp$income >  incQuants[2] & dlexp$income <= incQuants[3]] <- "25-50"
  dlexp$incomeQuant[dlexp$income >  incQuants[3] & dlexp$income <= incQuants[4]] <- "50-75"
  dlexp$incomeQuant[dlexp$income >  incQuants[4]] <- "75-100"

  # classification from code_supp.R
  dlexp$educ4[dlexp$educ == 1 | dlexp$educ == 2] <- 1 # high school
  dlexp$educ4[dlexp$educ == 3 | dlexp$educ == 4] <- 2 # some college
  dlexp$educ4[dlexp$educ == 5] <- 3 # 4-year college degree
  dlexp$educ4[dlexp$educ == 6] <- 4 # graduate degree

  dlexp$educationLevel[dlexp$educ == 1 | dlexp$educ == 2] <- "High School" # high school
  dlexp$educationLevel[dlexp$educ == 3 | dlexp$educ == 4] <- "Some College Experience"
  dlexp$educationLevel[dlexp$educ == 5] <- "4-Year College Degree"
  dlexp$educationLevel[dlexp$educ == 6] <- "Graduate Degree"

  dlexp$educLevel = as.factor(dlexp$educ4)

  dlexp$pol_party <- NULL
  dlexp$pol_party[dlexp$pid3 == 1] <- "Democrat"
  dlexp$pol_party[dlexp$pid3 == 2] <- "Independent"
  dlexp$pol_party[dlexp$pid3 == 3] <- "Republican"

  dlexp$Z = dlexp$video
  dlexp$Y = dlexp$response

  return (dlexp)
}
```

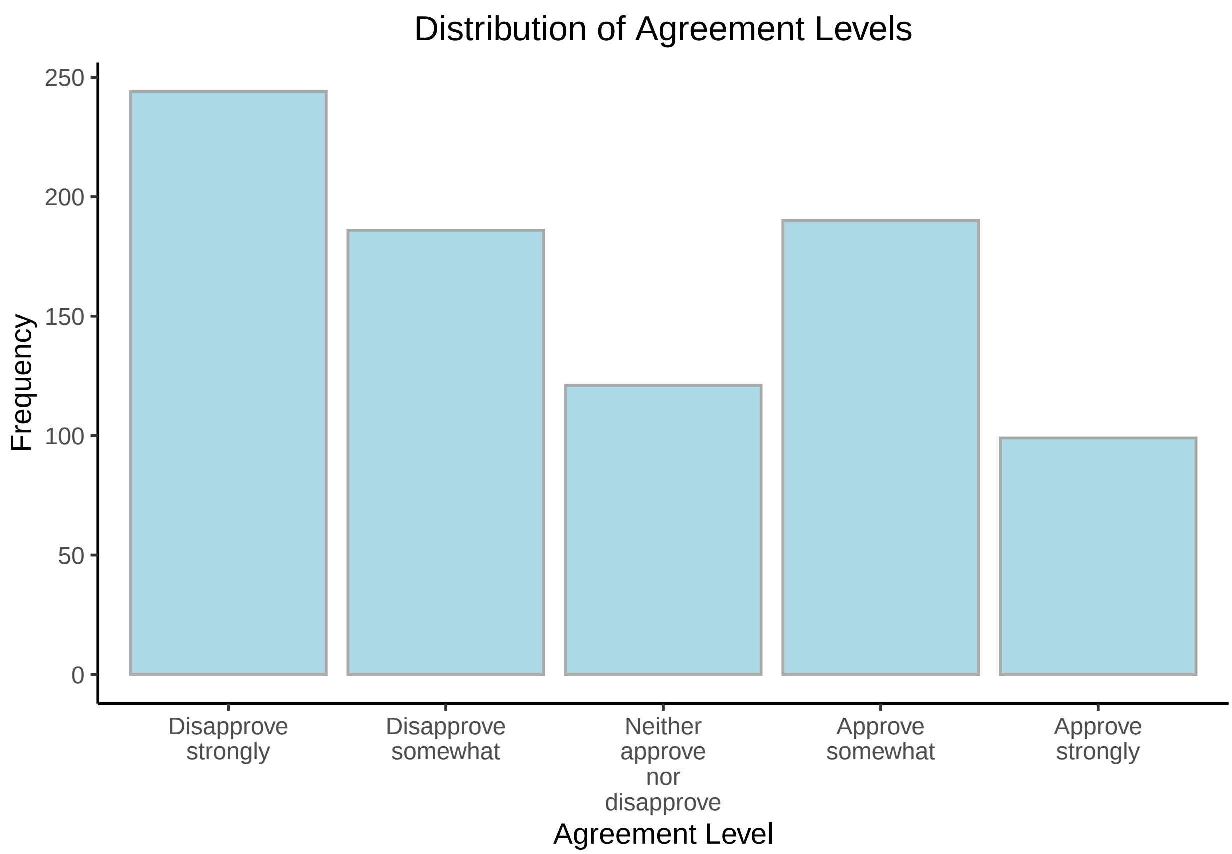
EDA.Rmd

Andri Vidarsson, Eden Luvishis, Veronica Yakubovich

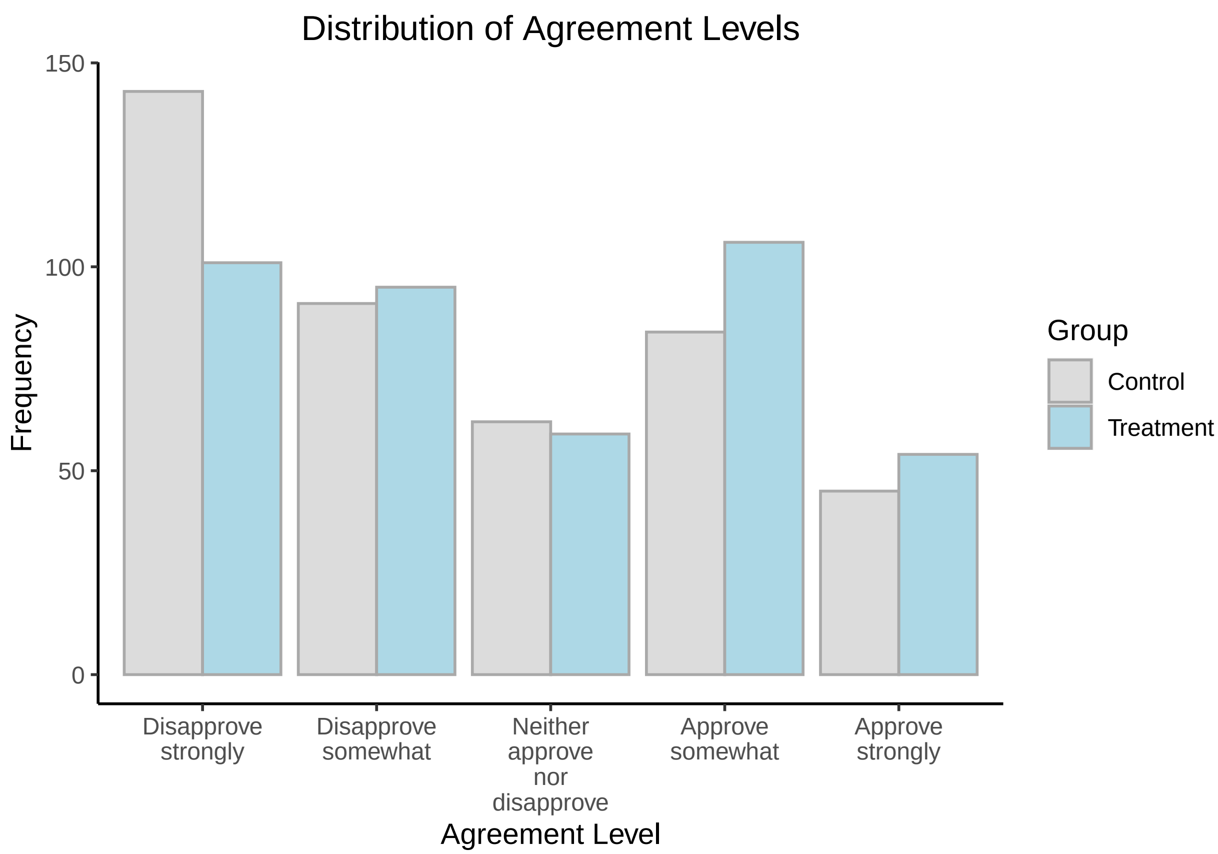
```r
library(dplyr)
library(ggplot2)
library(tidyr)
library(corrplot)
library(stringr)
source('dataProcessing.R')
```

```r
dlexp = loadData()

dlexp$agree = factor(dlexp$agree, levels = c("Disapprove strongly", "Disapprove
↪  somewhat", "Neither approve nor disapprove", "Approve somewhat", "Approve strongly"),
↪  ordered = TRUE)
```

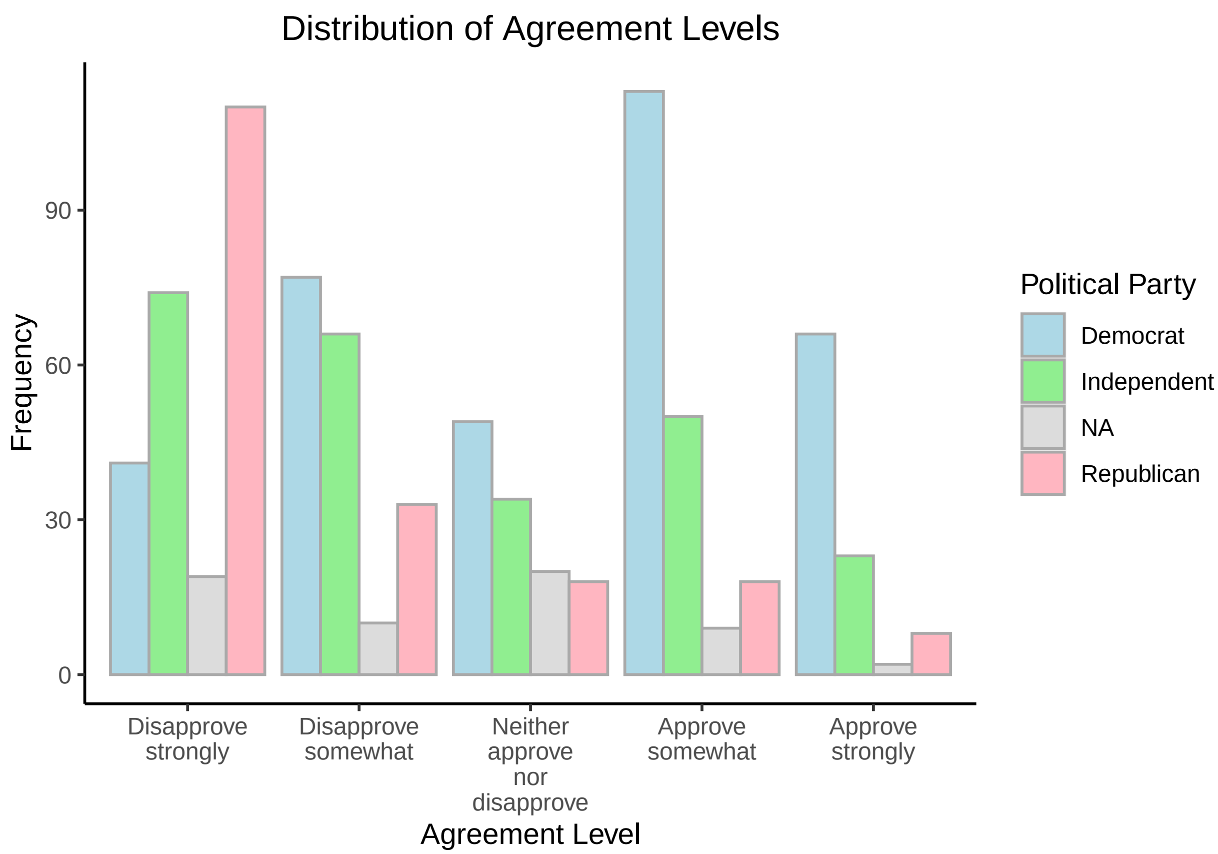
```r
df <- dlexp[dlexp$party.cues == 1,]

ggplot(df, aes(x = agree)) +
  geom_bar(fill = "lightblue", color = "darkgrey") +
  labs(title = "Distribution of Agreement Levels",
       x = "Agreement Level",
       y = "Frequency") +

  theme(plot.title = element_text(hjust = 0.5),
        panel.background = element_rect(fill = "white"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    text = element_text(color = "black")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10))
```




```
ggplot(df, aes(x = agree, fill = factor(Z))) +
  geom_bar(position = "dodge", color = "darkgrey") +
  labs(title = "Distribution of Agreement Levels",
       x = "Agreement Level",
       y = "Frequency") +

  theme(plot.title = element_text(hjust = 0.5),
        panel.background = element_rect(fill = "white"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    text = element_text(color = "black")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
scale_fill_manual(values = c("0" =  "#DCDCDC", "1" ="lightblue"), labels = c("0" =
↪  "Control", "1" = "Treatment"), name = "Group")
```

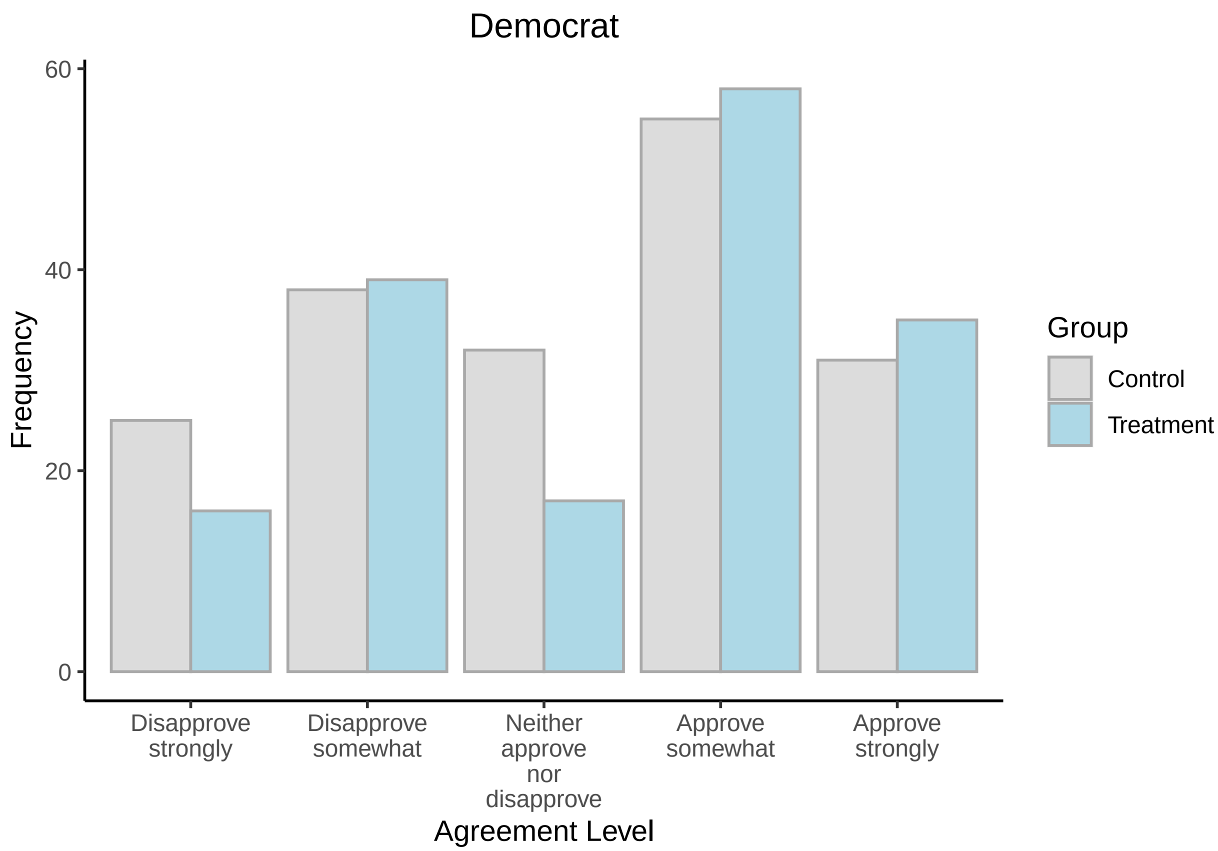



```
df$pol_party <- ifelse(is.na(df$pol_party), "NA", df$pol_party)

ggplot(df, aes(x = agree, fill = pol_party)) +
  geom_bar(position = "dodge", width = 0.9, colour = "darkgrey") +
  labs(title = "Distribution of Agreement Levels",
       x = "Agreement Level",
       y = "Frequency",
       fill = "Political Party") +
  theme(plot.title = element_text(hjust = 0.5),
        panel.background = element_rect(fill = "white"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(color = "black"),
    text = element_text(color = "black")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  scale_fill_manual(values = c("Democrat" = "lightblue", "Republican" = "#FFB6C1",
  ↪  "Independent" = "#90EE90", "NA" = "#DCDCDC"))
```

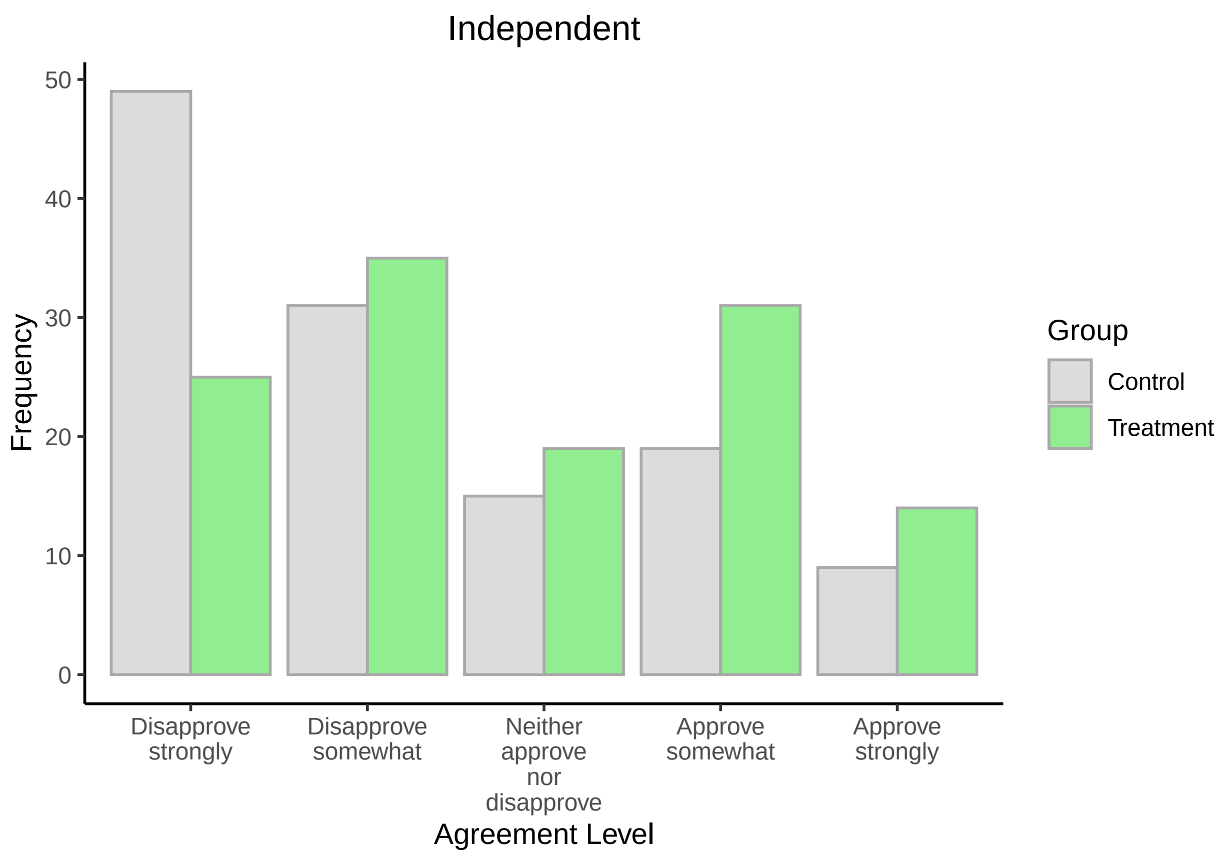



```
party_names = c('1' = "Democrat", '2' = "Independent", '3' = "Republican")
party_colors = c('1' = "lightblue", '2' = "#90EE90", '3' = "#FFB6C1")
dat = dlexp %>%
   filter(party.cues == 1) %>%
   group_by(across(all_of(c("pid3", "Z", "agree")))) %>%
   summarize(Frequency = n())

dat$Z = as.factor(dat$Z)

# barplot showing responses for treatment/control for selected political party
party_bar = function(pid){
  dat_ = filter(dat, pid3 == pid)
  ggplot(data=dat_, aes(x=agree, y=Frequency, fill=Z)) +
    geom_bar(stat="identity",  position="dodge", colour = "darkgrey") +
    ↪  ggtitle(party_names[pid]) +
    labs(x = "Agreement Level", y = "Frequency") +
    theme(plot.title = element_text(hjust = 0.5),
       panel.background = element_rect(fill = "white"),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
       axis.line = element_line(color = "black"),
       text = element_text(color = "black")) +
    scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
    scale_fill_manual(values = c("0" =  "#DCDCDC", "1" = party_colors[[pid]]), labels =
    ↪  c("0" = "Control", "1" = "Treatment"), name = "Group")
```

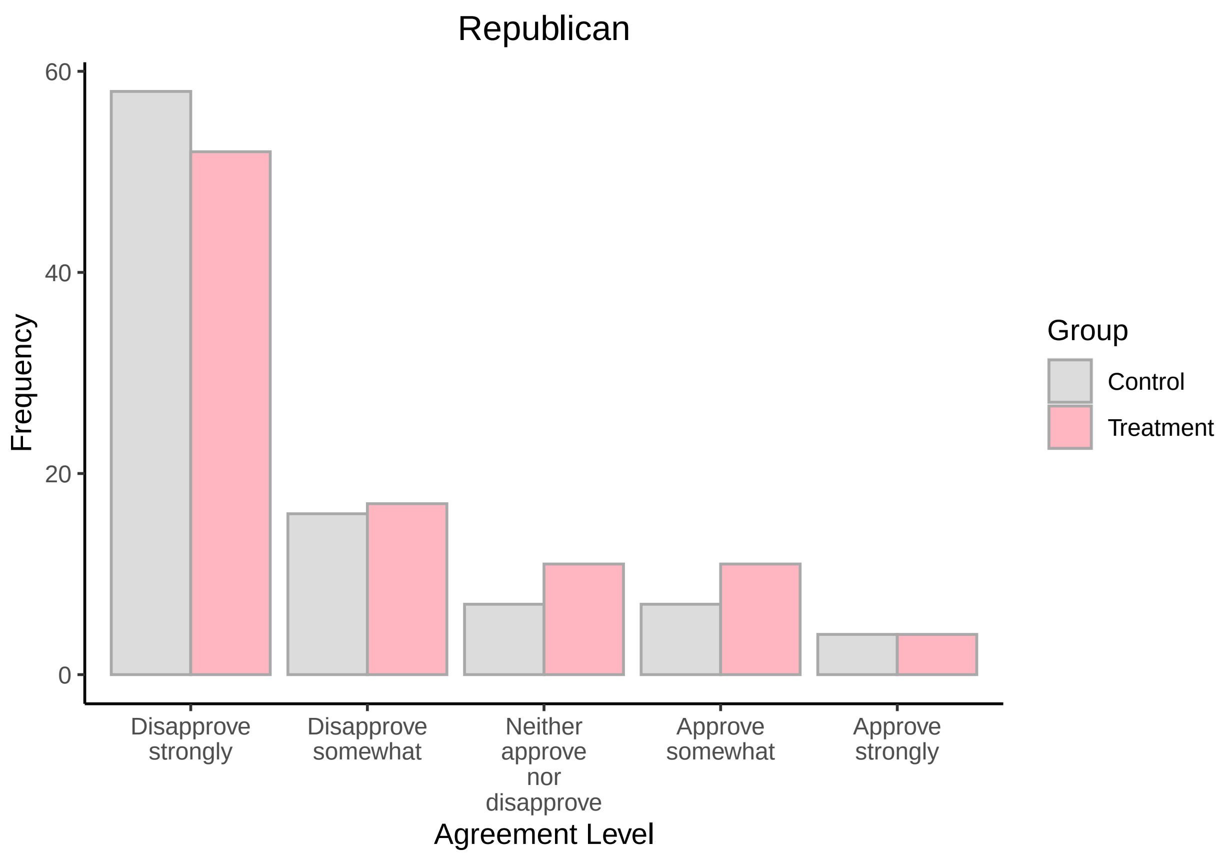
```
}
party_bar(1)
```




```
party_bar(2)
```

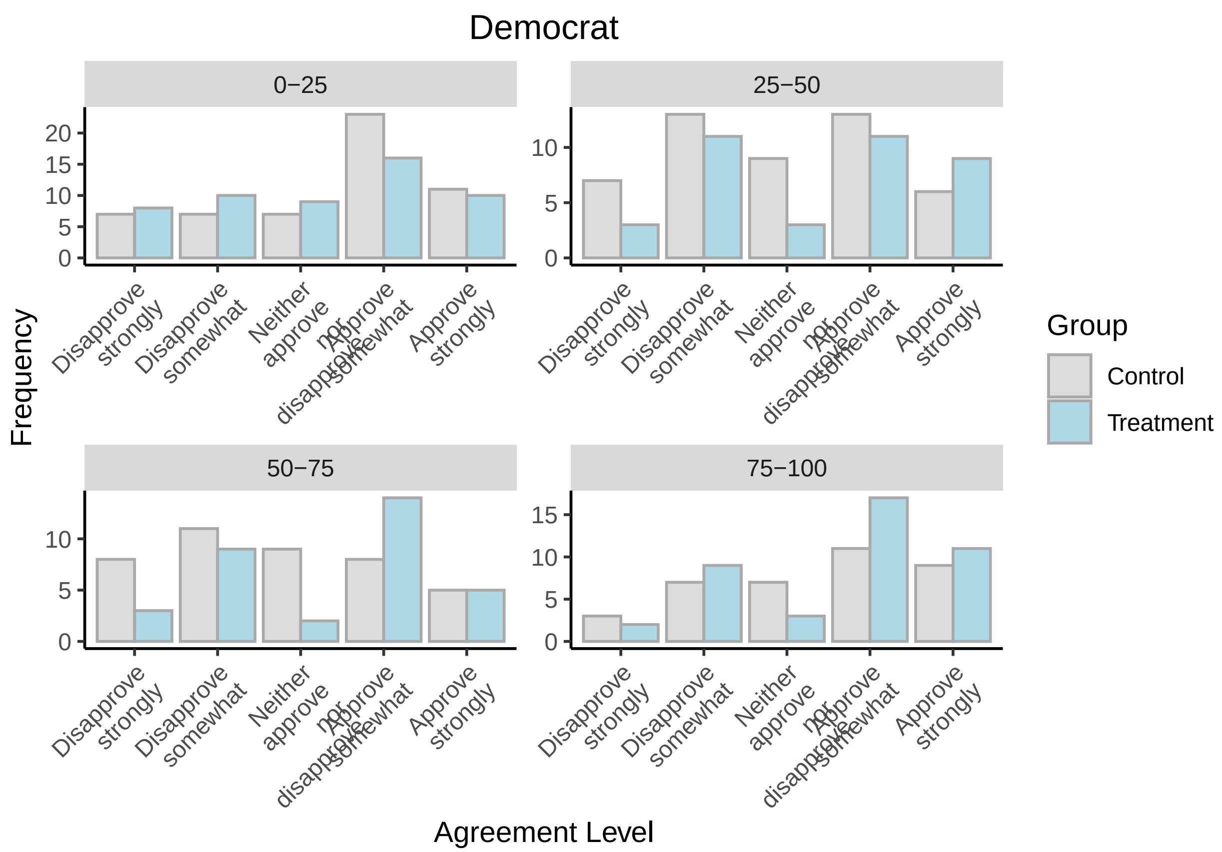

Independent
50
40
30
20
10
0
Frequency
Disapprove strongly
Disapprove somewhat
Neither approve nor disapprove
Approve somewhat
Approve strongly
Agreement Level
Group
Control
Treatment

```r
party_bar(3)
```

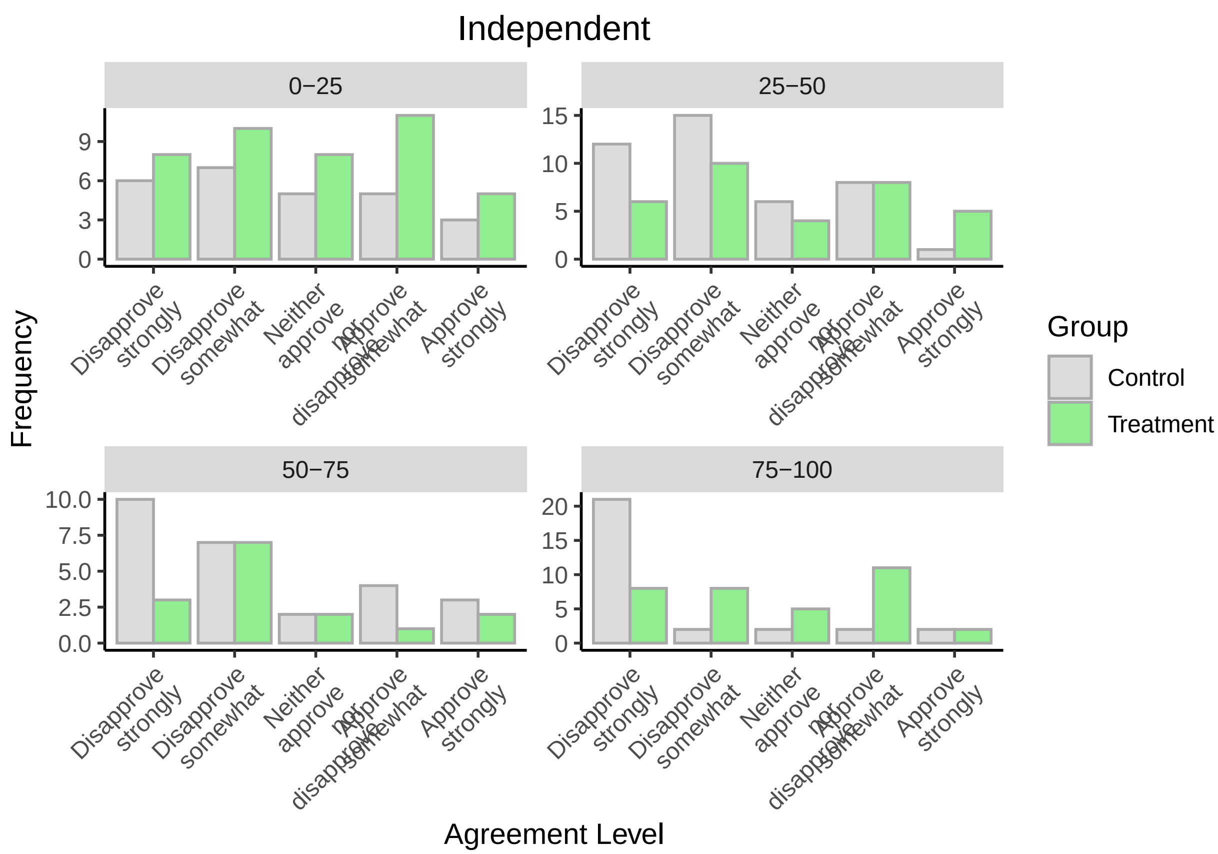



```
datInc = dlexp %>%
   filter(party.cues == 1) %>%
   group_by(across(all_of(c("pid3", "incomeQuant", "Z", "agree")))) %>%
   summarize(Frequency = n())

datInc$Z = as.factor(datInc$Z)

# show breakdown for responses by income quantile, per group
income_plot = function(pid){
  dat = filter(datInc, pid3 == pid)
  ggplot(data=dat, aes(x=agree, y=Frequency, fill=Z)) +
    geom_bar(stat="identity",  position="dodge", colour = "darkgrey") +
    facet_wrap(~incomeQuant, scales = "free") +
    ggtitle(party_names[pid]) +
    labs(x = "Agreement Level", y = "Frequency") +
    theme(plot.title = element_text(hjust = 0.5),
       panel.background = element_rect(fill = "white"),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
       axis.line = element_line(color = "black"),
       text = element_text(color = "black"),
       axis.text.x = element_text(angle = 45, hjust = 1)) +
    scale_x_discrete(labels = function(x) str_wrap(x, width = 1)) +
    scale_fill_manual(values = c("0" =  "#DCDCDC", "1" = party_colors[[pid]]), labels =
    ↪  c("0" = "Control", "1" = "Treatment"), name = "Group")
```

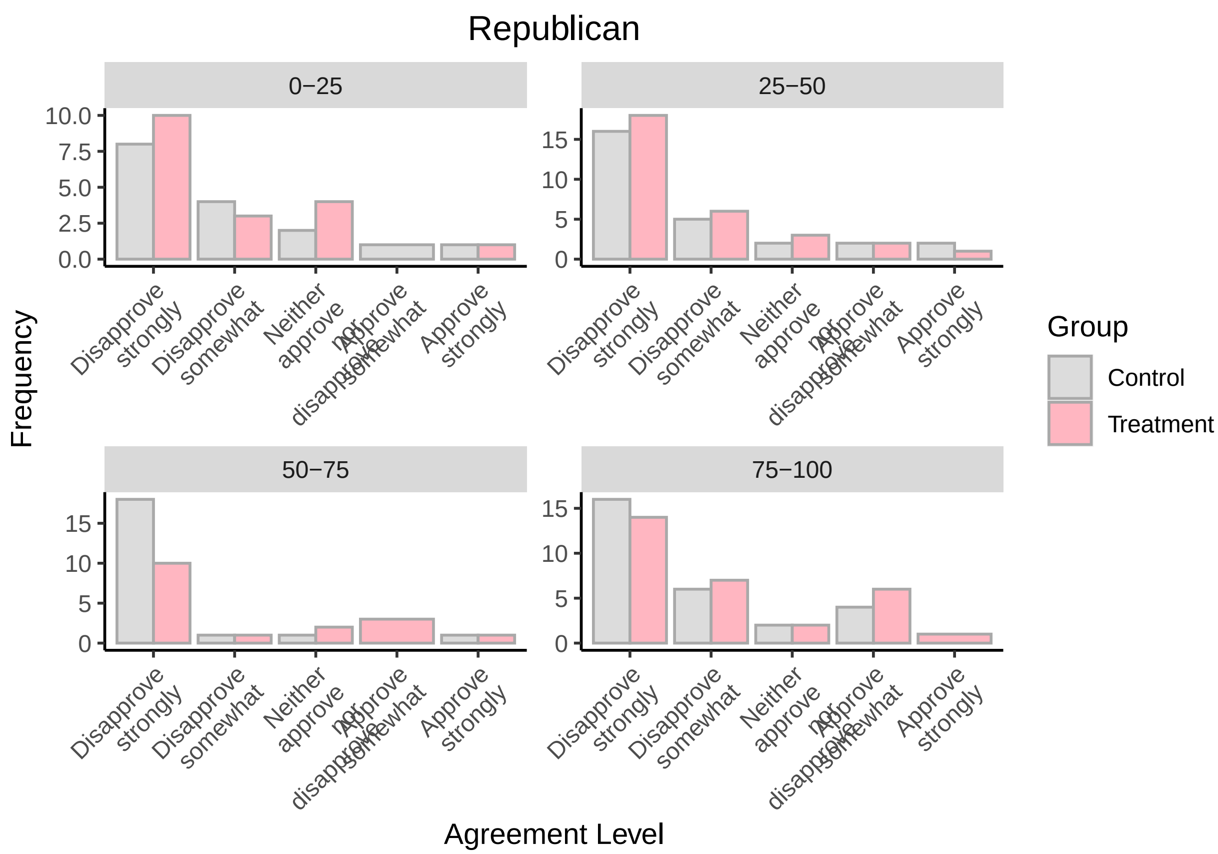
```r
}

income_plot(1)
```




```r
income_plot(2)
```

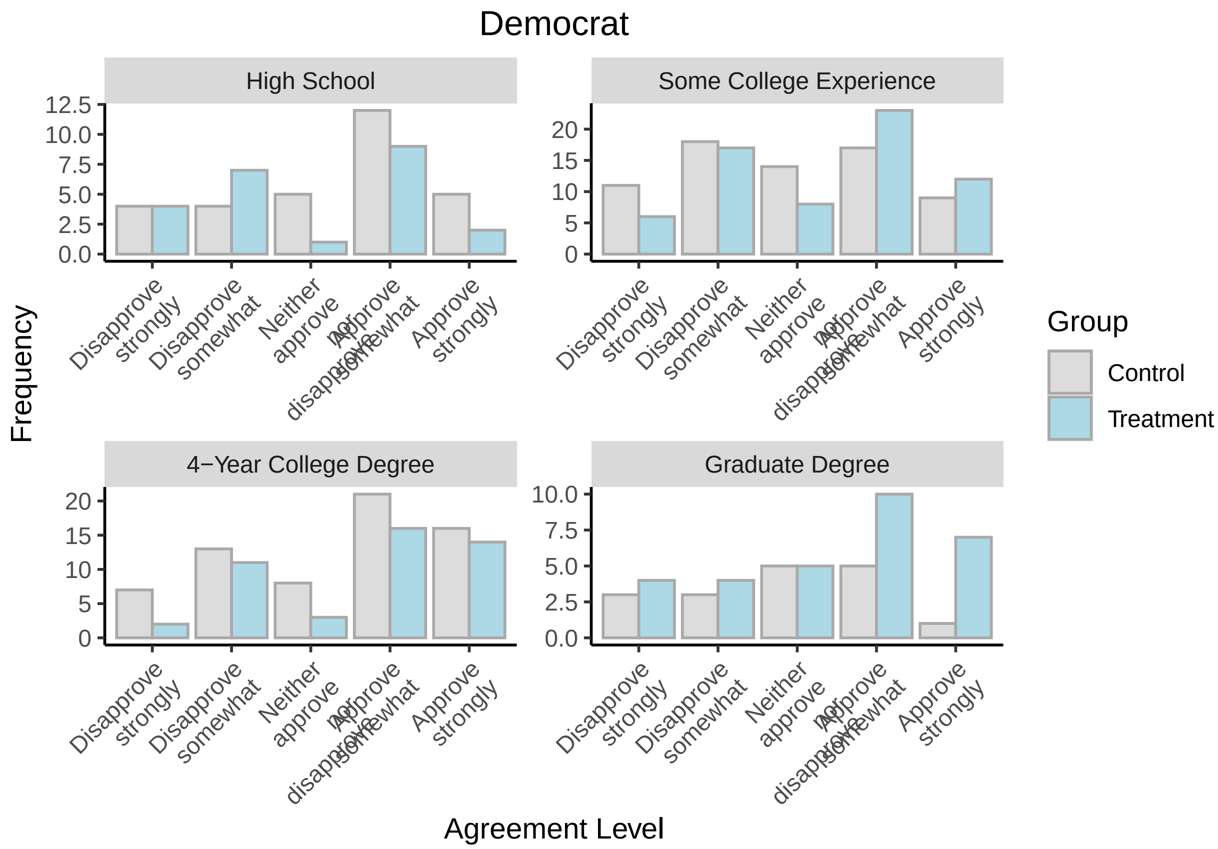



```r
income_plot(3)
```

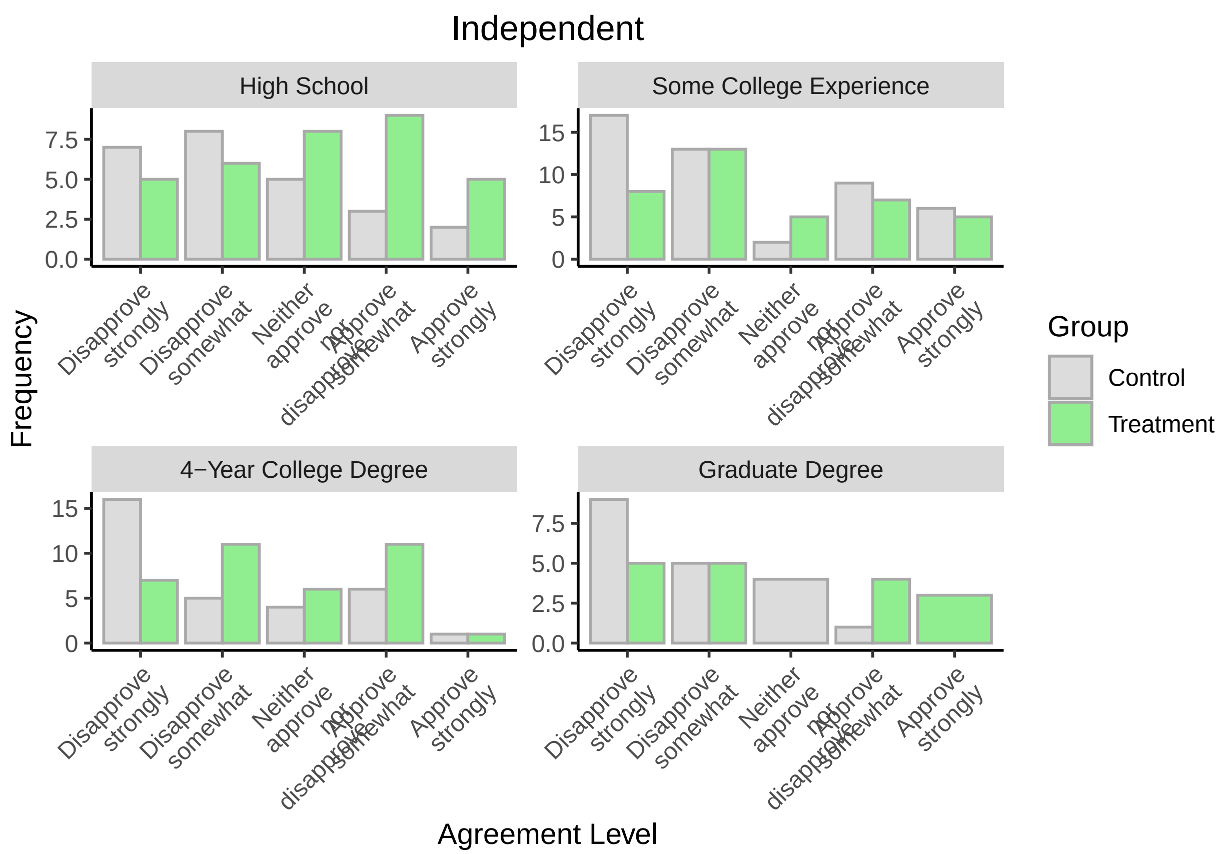



```
datEduc = dlexp %>%
   filter(party.cues == 1) %>%
   group_by(across(all_of(c("pid3", "educationLevel", "Z", "agree")))) %>%
   summarize(Frequency = n())

datEduc$Z = as.factor(datEduc$Z)

# show breakdown for responses by education level, per group
educ_plot = function(pid){
  dat = filter(datEduc, pid3 == pid)

  # maintain order of education - high school, college, ....
  dat = transform(dat,
      educationLevel=factor(educationLevel,
                            levels=c("High School","Some College Experience","4-Year
                            ↪  College Degree", "Graduate Degree")))

  ggplot(data = dat,
      aes(x=agree, y=Frequency, fill=Z)) +
    geom_bar(stat="identity",  position="dodge", colour = "darkgrey") +
    facet_wrap(~educationLevel, scales = "free") +
    ggtitle(party_names[pid]) +
    labs(x = "Agreement Level", y = "Frequency") +
    theme(plot.title = element_text(hjust = 0.5),
      panel.background = element_rect(fill = "white"),
      panel.grid.major = element_blank(),
```

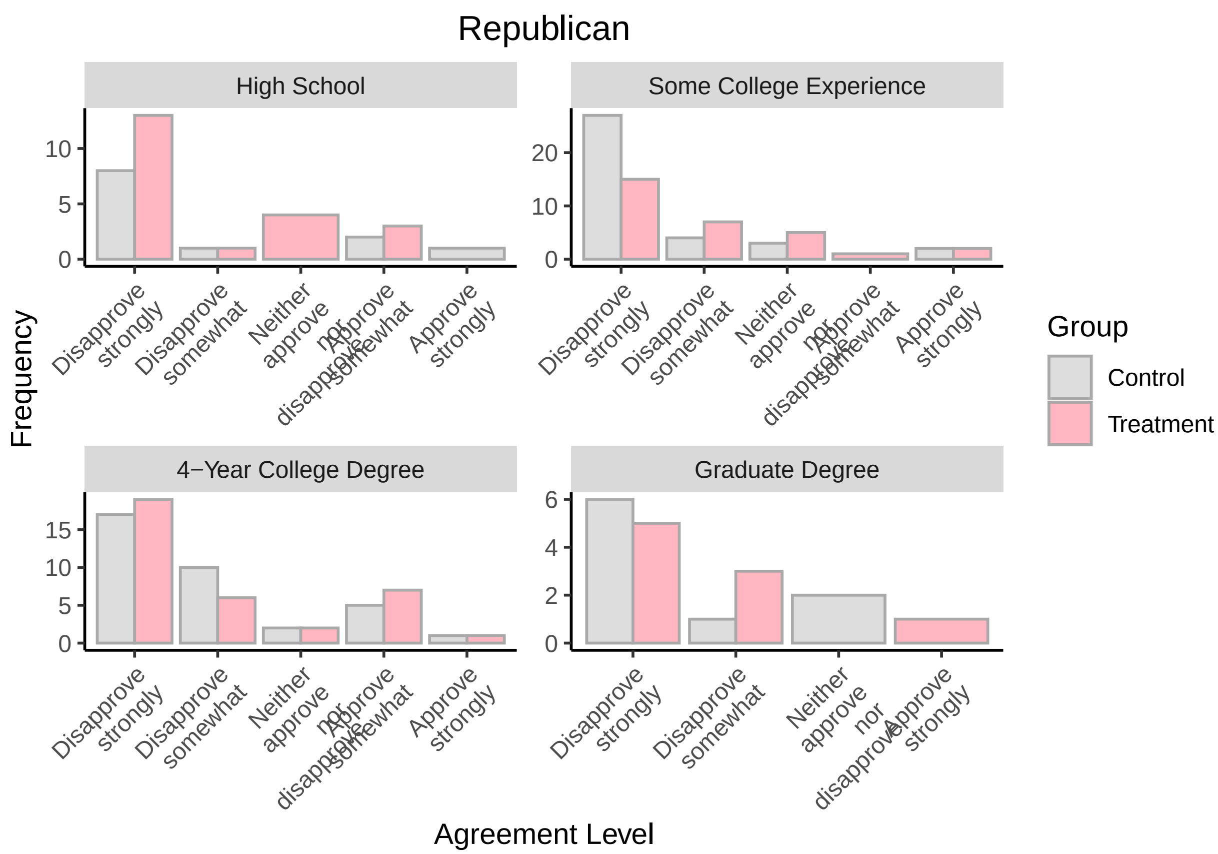
```
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "black"),
        text = element_text(color = "black"),
        axis.text.x = element_text(angle = 45, hjust = 1)) +
    scale_x_discrete(labels = function(x) str_wrap(x, width = 1)) +
    scale_fill_manual(values = c("0" =  "#DCDCDC", "1" = party_colors[[pid]]), labels =
    ↪  c("0" = "Control", "1" = "Treatment"), name = "Group")

}

educ_plot(1)
```




```
educ_plot(2)
```

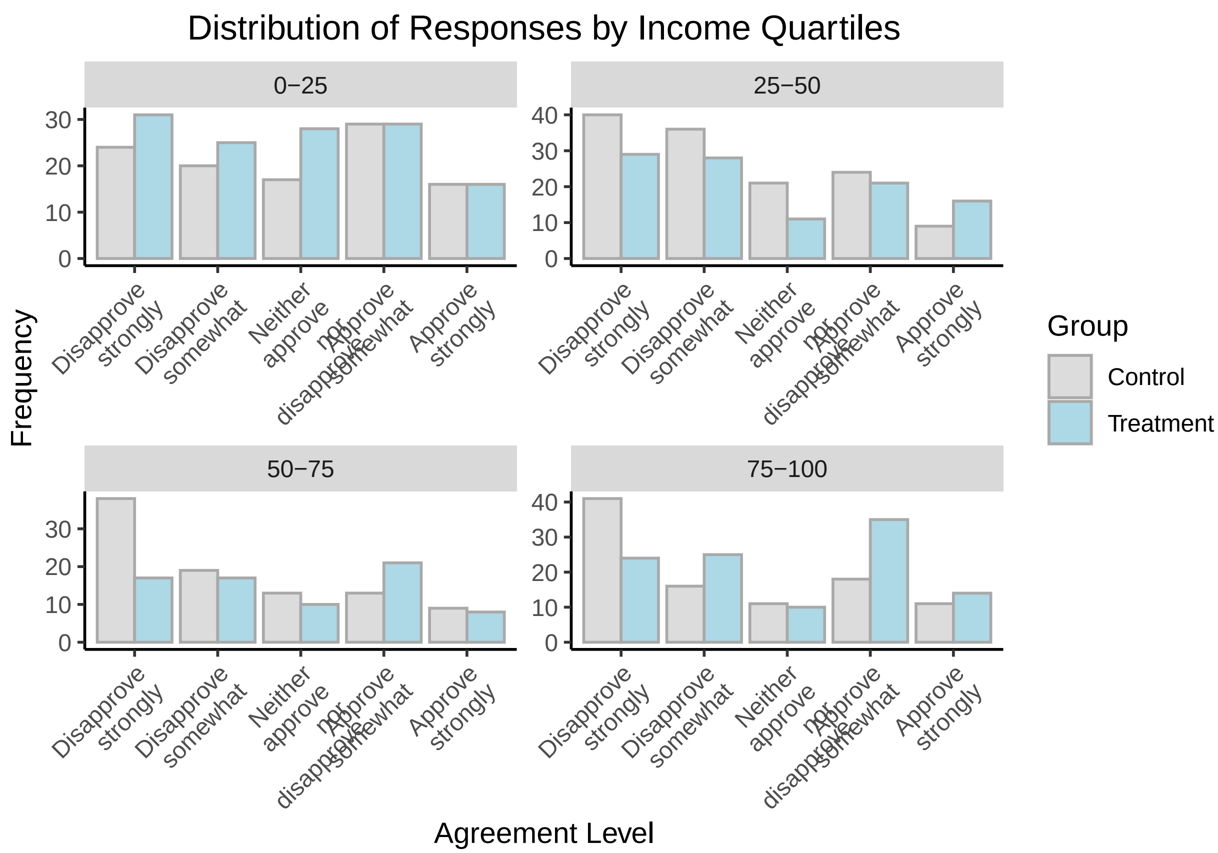

Independent
High School
Some College Experience
4−Year College Degree
Graduate Degree
Frequency
Agreement Level
Group
Control
Treatment
Disapprove strongly
Disapprove somewhat
Neither approve nor disapprove
Approve somewhat
Approve strongly

```r
educ_plot(3)
```




```
income_plot_ = function(){
  datInc_ = dlexp %>%
   filter(party.cues == 1) %>%
   group_by(across(all_of(c("incomeQuant", "Z", "agree")))) %>%
   summarize(Frequency = n())

  datInc_$Z = as.factor(datInc_$Z)

  ggplot(data=datInc_, aes(x=agree, y=Frequency, fill=Z)) +
    geom_bar(stat="identity",  position="dodge", colour = "darkgrey") +
    facet_wrap(~incomeQuant, scales = "free") +
    ggtitle("Distribution of Responses by Income Quartiles") +
    labs(x = "Agreement Level", y = "Frequency") +
    theme(plot.title = element_text(hjust = 0.5),
       panel.background = element_rect(fill = "white"),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
       axis.line = element_line(color = "black"),
       text = element_text(color = "black"),
       axis.text.x = element_text(angle = 45, hjust = 1)) +
    scale_x_discrete(labels = function(x) str_wrap(x, width = 1)) +
    scale_fill_manual(values = c("0" =  "#DCDCDC", "1" = "lightblue"), labels = c("0" =
    ↪  "Control", "1" = "Treatment"), name = "Group")
}
income_plot_()
```

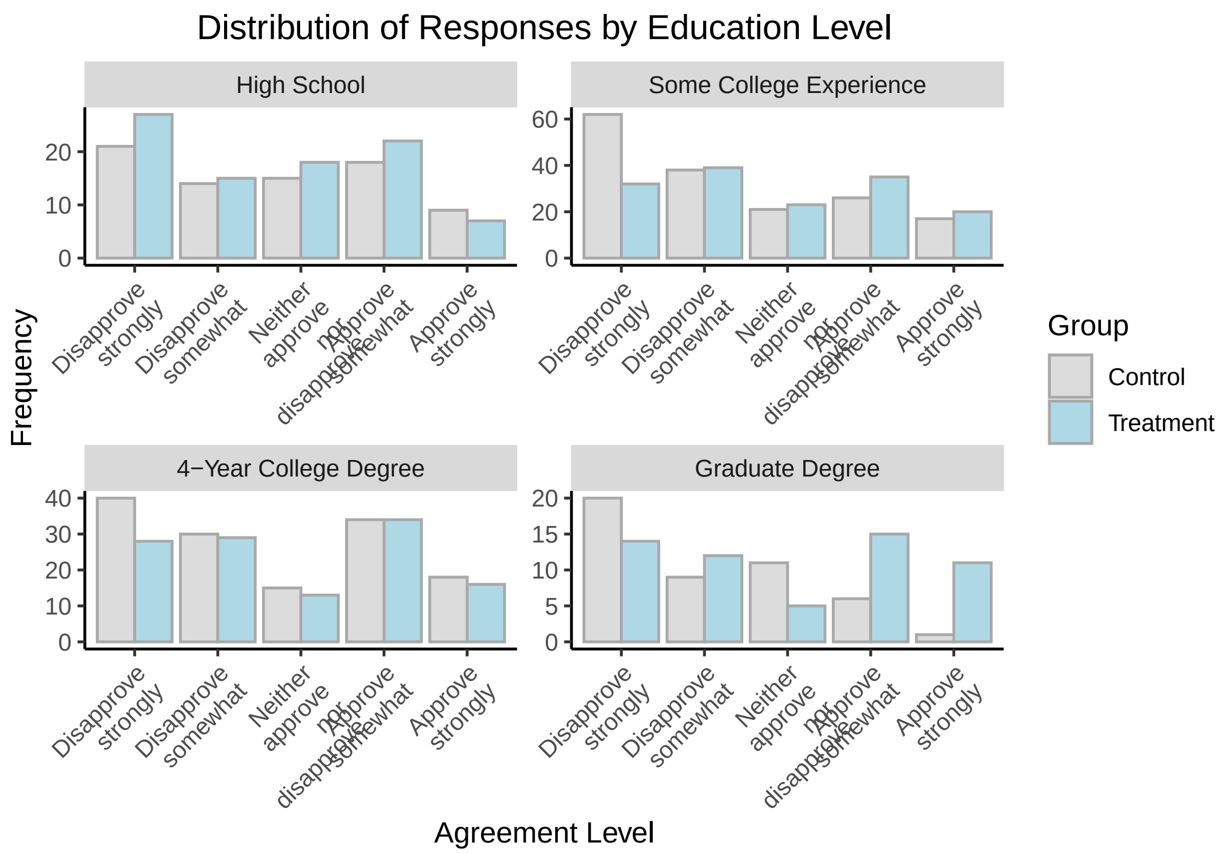
```
## `summarise()` has grouped output by 'incomeQuant', 'Z'. You can override using
## the `.groups` argument.
```



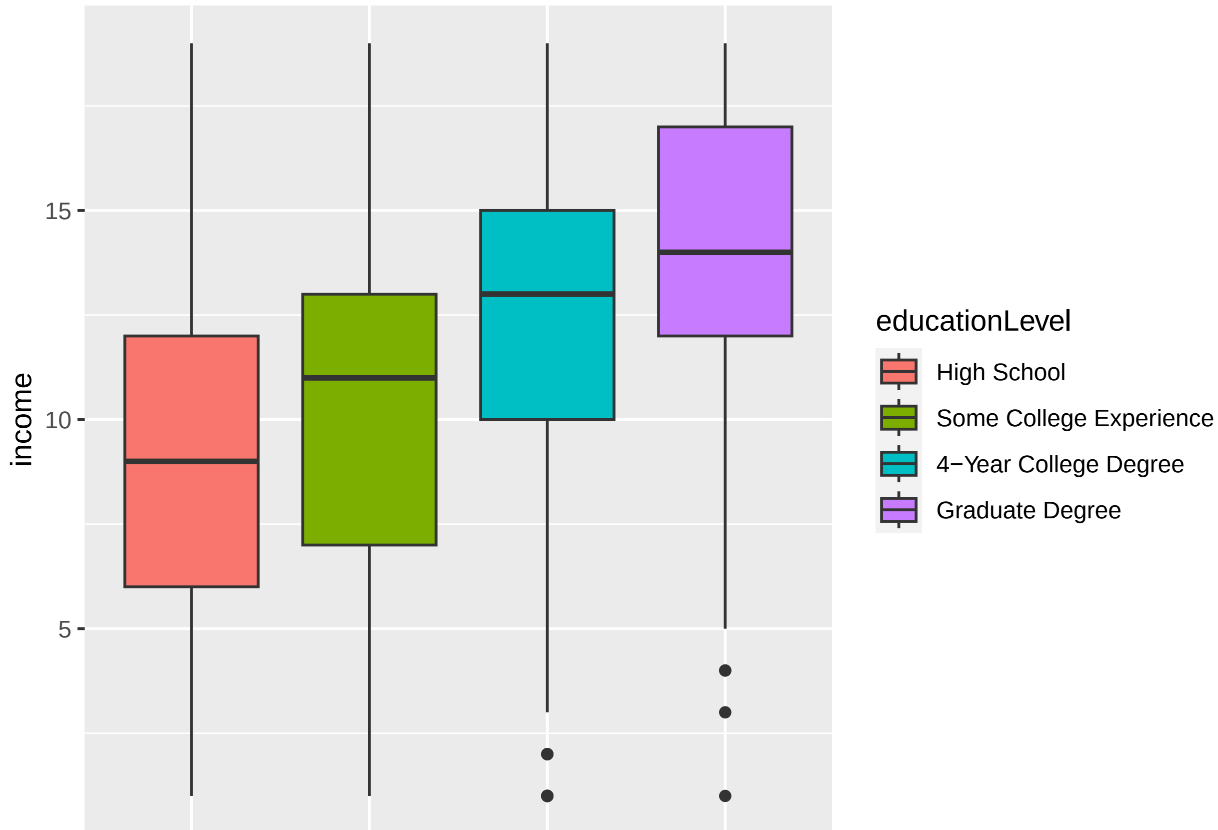

```
educ_plot_ = function(){
  educ_ = dlexp %>%
   filter(party.cues == 1) %>%
   group_by(across(all_of(c("educationLevel", "Z", "agree")))) %>%
   summarize(Frequency = n())

  educ_ = transform(educ_,
      educationLevel=factor(educationLevel,
                            levels=c("High School","Some College Experience","4-Year
                            ↪  College Degree", "Graduate Degree")))

  educ_$Z = as.factor(educ_$Z)

  ggplot(data=educ_, aes(x=agree, y=Frequency, fill=Z)) +
    geom_bar(stat="identity",  position="dodge", colour = "darkgrey") +
    facet_wrap(~educationLevel, scales = "free") +
    ggtitle("Distribution of Responses by Education Level")+
    labs(x = "Agreement Level", y = "Frequency") +
    theme(plot.title = element_text(hjust = 0.5),
       panel.background = element_rect(fill = "white"),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank(),
```

```r
    axis.line = element_line(color = "black"),
    text = element_text(color = "black"),
    axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 1)) +
  scale_fill_manual(values = c("0" =  "#DCDCDC", "1" = "lightblue"), labels = c("0" =
  ↪  "Control", "1" = "Treatment"), name = "Group")

}


educ_plot_()
```

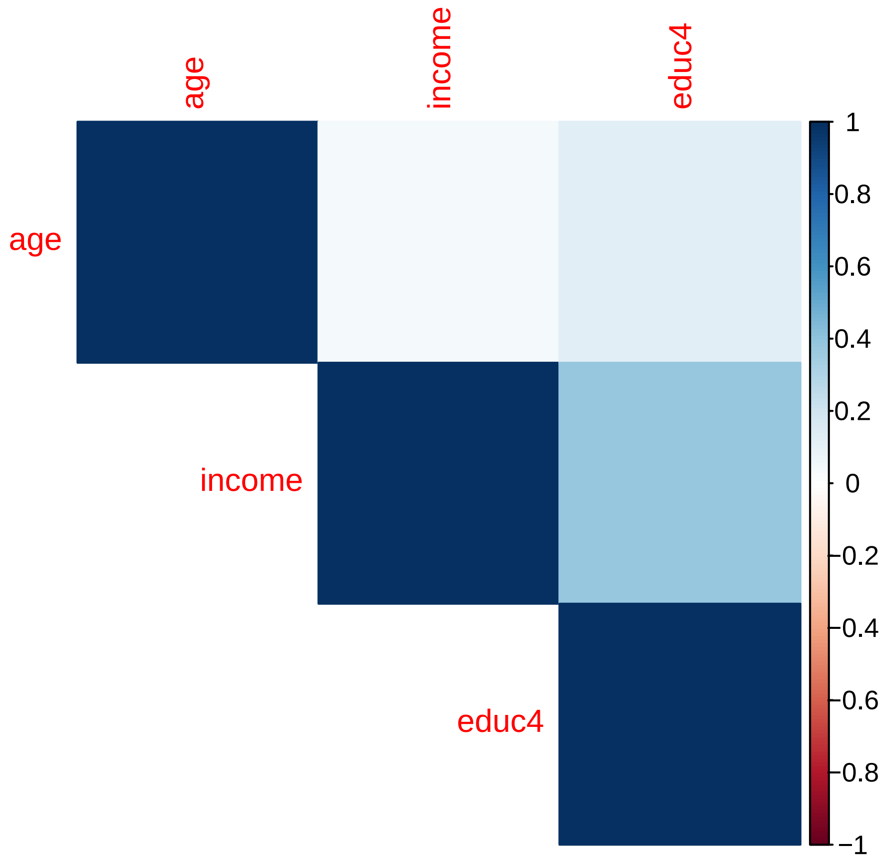
```
## `summarise()` has grouped output by 'educationLevel', 'Z'. You can override
## using the `.groups` argument.
```




```r
# Education-Income Correlation

dat = transform(dlexp,
      educationLevel=factor(educationLevel,levels=c("High School","Some College
      ↪  Experience","4-Year College Degree", "Graduate Degree")))
ggplot(aes(y = income, x = educationLevel), data = dat) +
geom_boxplot(aes(fill=educationLevel)) + theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```




```
dat_sub = dlexp[, c("income", "educ4", "age")]
cor_matrix <- cor(dat_sub)

corrplot(cor_matrix, method = "color", type = "upper", order = "hclust")
```


age
income
educ4
age
income
educ4
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Inference.Rmd

Andri Vidarsson, Eden Luvishis, Veronica Yakubovich

```r
library(dplyr)
source('dataProcessing.R')
library(estimatr)
library(randomForest)
library(latex2exp)
library(ggplot2)
dlexp = loadData()
```

```r
# Difference in means
meanDiff <- function(z, y){
  ZY = data.frame(z)
  ZY$y = y
  treat = filter(ZY, z == 1)
  ctrl = filter(ZY, z == 0)

  return (mean(treat$y) - mean(ctrl$y))
}

# FRT
frtSim <- function(z, y, M, testStat, tObserved){
  pVals = vector(, M) # for p-value convergence
  pValCnt = 0
  for (i in 1:M){
    set.seed(i)
    zSim = sample(z)
    t = testStat(zSim, y)
    if (t >= tObserved){
      pValCnt = pValCnt + 1
    }
    pVals[i] = pValCnt/i
  }

  return (pVals)
}
```

```r
# Neymanian inference
s_2_z <- function(data, z){
  sub_ = filter(data, Z == z)
  n_z = nrow(sub_)

  return (c(var(sub_$Y), n_z))
}
```

```r
# conservative variance estimate for Neyman
vHat = function(data){
  s_2_1 = s_2_z(data, 1)
  s_2_0 = s_2_z(data, 0)

  v_hat = (s_2_1[1]/s_2_1[2]) + (s_2_0[1]/s_2_0[2])
  return (v_hat)
}

# normal approx 95% confidence interval
ci <- function(that, vhat){
  return (c(that - 1.96*sqrt(vhat), that + 1.96*sqrt(vhat)))
}
```

```r
# discard those who did not get party cue
dParty = filter(dlexp, party.cues == 1, !is.na(pid3))

# run difference in means
tObs = meanDiff(dParty$Z, dParty$Y)
print(tObs)
```

```
## [1] 0.2696496
```

```r
# run frt
pVals = frtSim(dParty$Z, dParty$Y, 10000, meanDiff, tObs)
pVal = pVals[length(pVals)] # neg index does not work?
print(pVal)
```

```
## [1] 0.0041
```

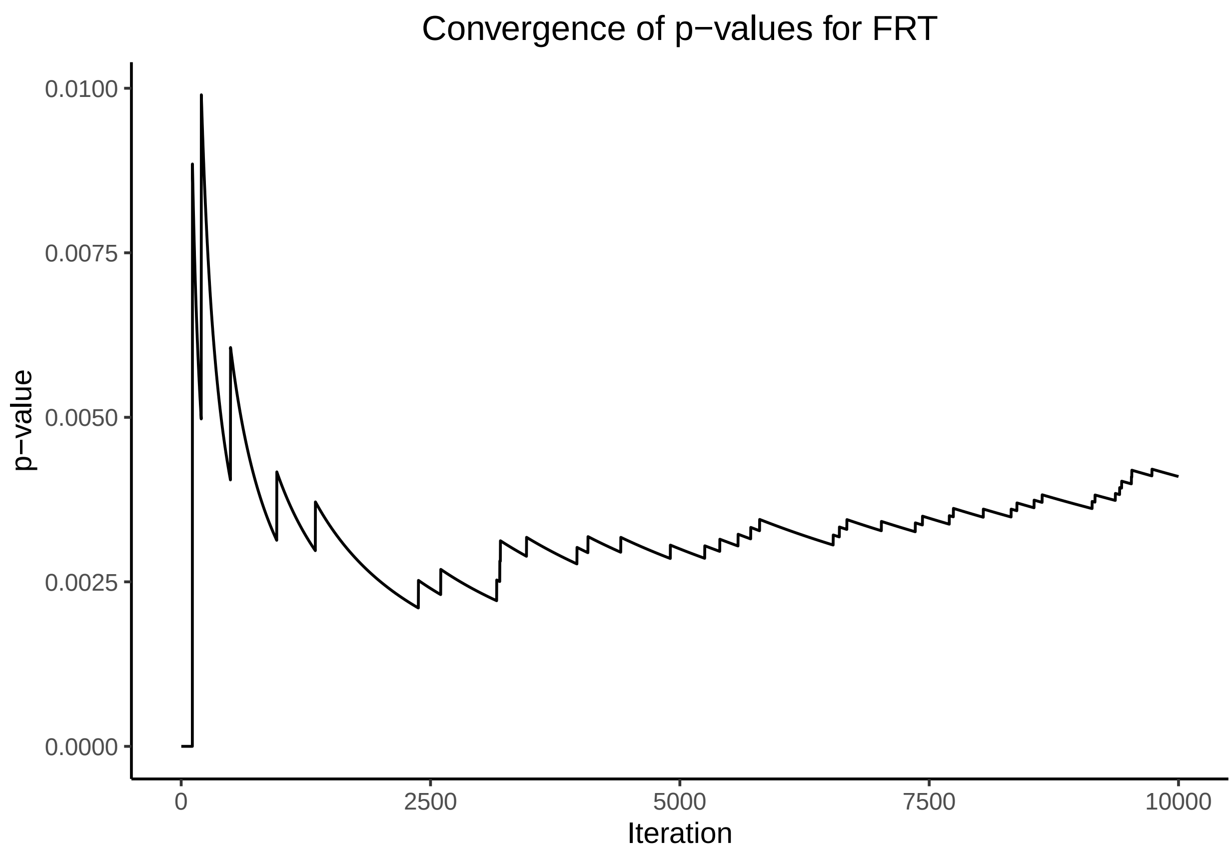
```r
# get neymanian variance estimate and confidence interval
v_hat = vHat(dParty)
print(v_hat)
```

```
## [1] 0.01021915
```

```r
print(ci(tObs, v_hat))
```

```
## [1] 0.0715136 0.4677856
```

```r
# plot p-value convergence for FRT
dfP <- data.frame(Index = seq_along(pVals), pVals = pVals)

ggplot(dfP, aes(x = Index, y = pVals)) +
  geom_line() +
  labs(x = "Iteration", y = "p-value", title = "Convergence of p-values for FRT") +
  theme(plot.title = element_text(hjust = 0.5),
       panel.background = element_rect(fill = "white"),
       axis.line = element_line(color = "black"),
       text = element_text(color = "black"))
```




```
# run neymanian inference for each party/group separately

runParty = function(data, pid){
  dat = filter(data, pid3 == pid)
  tau_hat = meanDiff(dat$Z, dat$Y)
  v_hat = vHat(dat)
  ci_ = ci(tau_hat, v_hat)
  return (c(tau_hat, ci_))
}

# plot confidence intervals for each party/group on same plot
confIntPlot = function(dem, ind, rep, est_expr){
  df = rbind(dem, ind, rep)
  df <- as.data.frame(df)

  new_column_names <- c("tau_hat", "ci_lower", "ci_upper")
  colnames(df) <- new_column_names
  df$politicalParty = c("Democrat", "Independent", "Republican")

  ggplot(df, aes(politicalParty, tau_hat)) + geom_point() +
    geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper)) +
    labs(x = "Political Party", y = TeX(est_expr), title = "Confidence Intervals for Each
    ↪  Party") +
    coord_flip() +
    theme(plot.title = element_text(hjust = 0.5),
       panel.background = element_rect(fill = "white"),
```

```
       panel.grid.major = element_line(color = "#F5F5F5"),
       panel.grid.minor = element_line(color = "#F5F5F5"),
       axis.line = element_line(color = "black"),
       text = element_text(color = "black"))
}

(neymDem = runParty(dParty, 1)) # dem
```

```
## [1]  0.18523355 -0.09183471  0.46230181
```

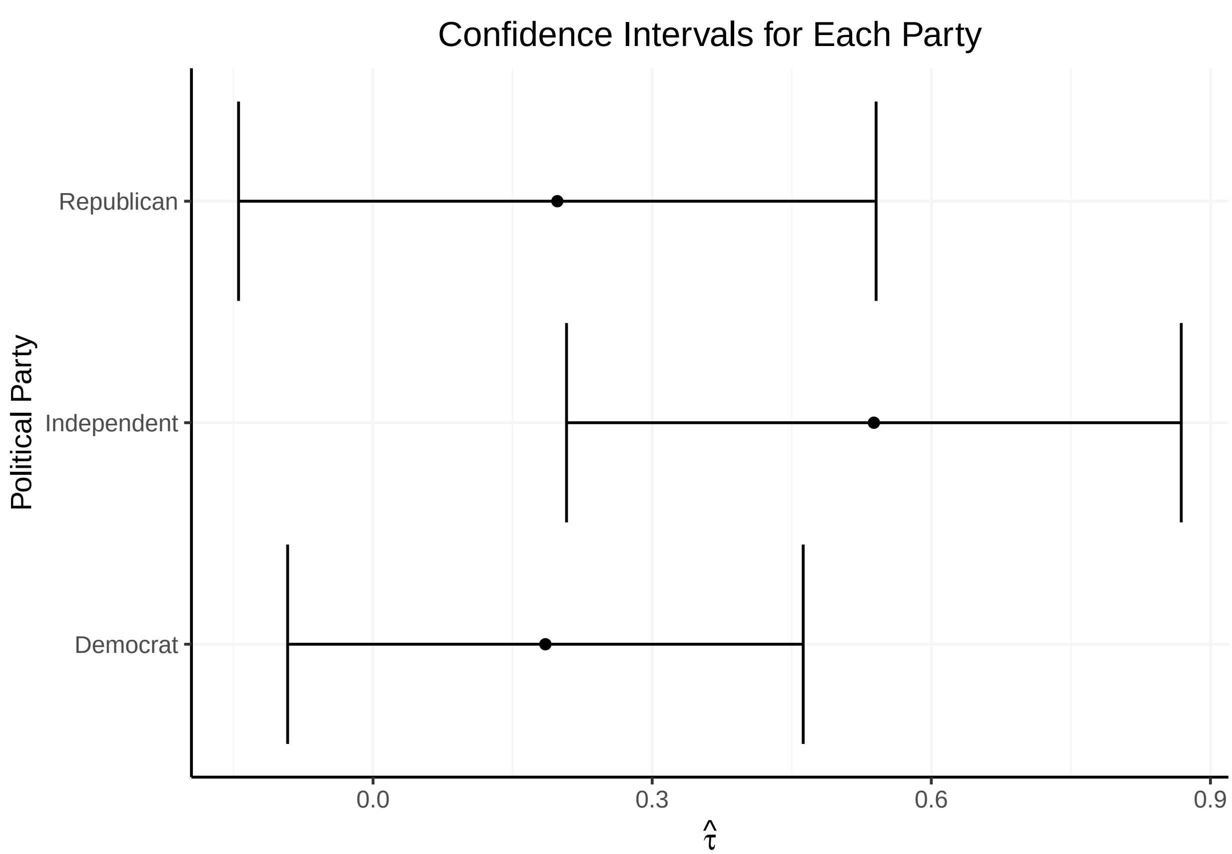
```
(neymInd = runParty(dParty, 2)) # ind
```

```
## [1] 0.5382901 0.2079570 0.8686231
```

```
(neymRep = runParty(dParty, 3)) # rep
```

```
## [1]  0.1980549 -0.1445401  0.5406500
```

```
# No surprises here, same results as paper


confIntPlot(neymDem, neymInd, neymRep, "$\\hat{\\tau}$")
```



```r
dParty$Y_factor = as.factor(dParty$Y)

# shift by centering of other data
centerVars = function(dataCenter, dataShift){
  vars = c("income", "educ4", "age", "white") # covariates included in model, to shift
  data_ = dataShift
  for (var in vars){
    data_[, var] <- data_[, var] - mean(dataCenter[, var])
  }
  return (data_)
}
```

```r
# Generalized Lin's Estimator with random forest model

# Using classification to predict and then interpret as continuous for variance te/var
↪  calc
trainModel = function(data){
  data = droplevels(data)
  model = randomForest(Y_factor ~ income + educ4 + age + white, data = data)
  return (model)
}

# predict for x
# training data: data that model was trained on
predict_calibrate_cross = function(model,  x, leftout_data, training_data){
  # shift left-out data based on centering for training data
  x_shifted = centerVars(training_data, x)
  leftout_data_shifted = centerVars(training_data, leftout_data)

  n_d = nrow(leftout_data)
  mu_hat = as.numeric(predict(model, x_shifted))
  l_pred = as.numeric(predict(model, leftout_data_shifted))

  l_pred_shift = sum((leftout_data$Y) - l_pred)/n_d

  mu_out = mu_hat + l_pred_shift
  return (mu_out)
}


# t_data: training data
# l_data: left-out data
tau_pred_cross = function(control_model, treatment_model,
                          control_t_data, treatment_t_data,
                          control_l_data, treatment_l_data){

  n_t = nrow(control_t_data) + nrow(treatment_t_data)

  tau = (sum(treatment_t_data$Y) +
           sum(predict_calibrate_cross(treatment_model, control_t_data, treatment_l_data,
           ↪  treatment_t_data)) -
           sum(control_t_data$Y) -
           sum(predict_calibrate_cross(control_model, treatment_t_data, control_l_data,
           ↪  control_t_data)))/n_t
```

```
  return (tau)
}

t_pred_cross_total = function(control_model_1, control_model_2,
                              treatment_model_1, treatment_model_2,
                              control_data_1, control_data_2,
                              treatment_data_1, treatment_data_2){

  n_1 = nrow(control_data_1) + nrow(treatment_data_1)
  n_2 = nrow(control_data_2) + nrow(treatment_data_2)
  n = n_1 + n_2

  t_out = (tau_pred_cross(control_model_2, treatment_model_2,
                          control_data_2, treatment_data_2,
                          control_data_1, treatment_data_1)*(n_1/n)) +
    (tau_pred_cross(control_model_1, treatment_model_1,
                    control_data_1, treatment_data_1,
                    control_data_2, treatment_data_2)*(n_2/n))

  return (t_out)
}


get_tau_cross = function(data){
  set.seed(0102)
  split_ind = sample(seq_len(nrow(data)), size = floor(0.5*nrow(data)))

  d1 = data[split_ind, ]
  d1_centered = centerVars(d1, d1)

  d2 = data[-split_ind, ]
  d2_centered = centerVars(d2, d2)

  d1_ctrl = filter(d1, Z == 0)
  d1_ctrl_centered = filter(d1_centered, Z == 0)
  d1_treatment = filter(d1, Z == 1)
  d1_treatment_centered = filter(d1_centered, Z == 1)

  d2_ctrl = filter(d2, Z == 0)
  d2_ctrl_centered = filter(d2_centered, Z == 0)
  d2_treatment = filter(d2, Z == 1)
  d2_treatment_centered = filter(d2_centered, Z == 1)

  ctrl_model_1 = trainModel(d1_ctrl_centered)
  treatment_model_1 = trainModel(d1_treatment_centered)

  ctrl_model_2 = trainModel(d2_ctrl_centered)
  treatment_model_2 = trainModel(d2_treatment_centered)


  t_pred = t_pred_cross_total(ctrl_model_1, ctrl_model_2,
                              treatment_model_1, treatment_model_2,
```

```r
                              d1_ctrl, d2_ctrl,
                              d1_treatment, d2_treatment)

  return (t_pred)

}

tau_hat_all = get_tau_cross(dParty)
print(tau_hat_all) # treatment effect estimate
```

```
## [1] 0.1876604
```

```r
# bootstrap variance, this takes a long time to run for generalized Lin's
boostrapTau = function(data, tau_function, M){
  t = vector(, M)
  for (i in 1:M){
    set.seed(i)
    data_sample = sample_n(data, nrow(data), replace = TRUE)
     t[i] = tau_function(data_sample)
  }
  return (t)
}
```

```r
bootSampleAll = boostrapTau(dParty, get_tau_cross, 1000)
```

```r
# confidence interval for treatment effect when running on entire dataset
ci(tau_hat_all, var(bootSampleAll))
```

```
## [1] -0.0498380  0.4251588
```

```r
# Run Lin's for each group and get confidence intervals for the treatment effect for each
↪  group
runLins = function(pid){
  data = filter(dParty, pid3 == pid)
  tau_est = get_tau_cross(data)
  bootSample = boostrapTau(data, get_tau_cross, 1000)

  ci_boot = ci(tau_est, var(bootSample))
  return (c(tau_est, ci_boot))
}

(l1 = runLins(1)) # dem
```

```
## [1]  0.1464305 -0.1822321  0.4750931
```

```r
(l2 = runLins(2)) # ind
```

```
## [1] 0.5045623 0.1111378 0.8979869
```

```r
(l3 = runLins(3)) # rep
```

```
## [1]  0.2538867 -0.1176119  0.6253853
```

```r
# plot confidence intervals for treatment effect from generalized Lin's estimator, per
↪  group
confIntPlot(l1, l2, l3, "$\\hat{\\tau}_L$")
```