# Final Project

Hugh Ford and Lauren Chandler-Holtz

2025-05-02

This project builds on analysis from Cheng and Hoekstra (2013) by re-examining whether strengthening Castle Doctrine laws leads to an increase in lethal violence. Castle Doctrine is a type of self-defense law that permits the use of deadly force to protect one's home. During the 2000s, many U.S. states rapidly adopted and expanded such laws. In our analysis, we investigate the causal impact of implementing these laws on states' murder rates from 2001 to 2010. We employ a comprehensive causal framework that includes constructing a directed acyclic graph (DAG), inverse probability weighting, G-computation, and sensitivity analyses. Our findings suggest that the adoption of Castle Doctrine laws does not have a statistically significant effect on murder rates relative to pre-enactment levels.

## Causal Question

Using data assembled by Cheng Cheng and Mark Hoekstra for their study, "Does Strengthening Self-Defense Law Deter Crime or Escalate Violence? Evidence from Expansions to Castle Doctrine" (Cheng and Hoekstra 2013), primarily sourced from the FBI Uniform Crime Reports Summary, we sought to examine the causal relationship between implementing Castle Doctrine and murder rates in the U.S. Focusing on the "treated" population—states that passed Castle Doctrine laws between 2001 and 2010, we investigated whether these states should repeal Castle Doctrine laws in order to bring down murder rates. Formally, our causal question was: among the states that passed Castle Doctrine laws between 2001 and 2010, what was the effect of that implementation on the state's murder rate?

# Background

As Cheng and Hoekstra explain in their work, “Castle Doctrine” stems from English Common Law, allowing an exception to the “duty to retreat” when an individual is in their own home (Cheng and Hoekstra 2013). In the United States, the legal history of Castle Doctrine dates back to the 1700s as a legacy of English colonial rule. During the era of Westward Expansion, the concept broadened to cover not only the home but also surrounding property and, in many cases, any place one had a legal right to be (Jay 2006). Castle Doctrine principles were formally codified in 1985, when Colorado passed the “Make My Day” law, which removed any civil or criminal liability for the use of force, including lethal force, against a home invader (Johnson 1990).

The modern expansion of self-defense laws began with Florida in 2005, when it became the first state to strengthen its self-defense protections and explicitly expand Castle Doctrine to places outside the home (Cheng and Hoekstra 2013). This change in statute was quickly adopted by about twenty other states in the following years. Since 2005, Castle Doctrine has spread to the significant majority of US states. However, laws vary state to state, with some being limited to the home while others are expanded to include other places, e.g., one’s place of work or in one’s vehicle.

Castle Doctrine often goes hand-in-hand with so-called “Stand Your Ground” laws; however, Castle Doctrine is distinguished from such laws in that it specifically applies to the home (and sometimes other areas such as one’s workplace or car) and allows the use of deadly force, even if disproportionate. In contrast, Stand Your Ground typically applies to anywhere one has a legal right to be but only allows defensive proportional force (Shouse 2025).

Proponents of Castle Doctrine argue that it is a useful deterrent to crime and contributes to public safety. Critics, on the other hand, contend that these laws may instead unnecessarily escalate levels of violence. The aim of our research is to investigate whether Castle Doctrine laws have a significant impact on states’ murder rates. For the purpose of our research, “murder rate” excludes cases that were ruled justifiable homicides.

# Creating a Directed Acyclic Graph (DAG)

We began by creating a DAG to represent the causal relationships between variables in our dataset. First, we specified our outcome as the state’s murder rate per 100,000 citizens and our exposure as passing a “Castle Doctrine” law between 2001 and 2010, as designated by Cheng and Hoekstra. We then identified several covariates we thought would play important causal roles in relation to murder rate and whether a state would pass a Castle Doctrine law.

For example, we believed that the murder rate as well as the decision to pass a Castle Doctrine Law could depend on the state's demographics as well as the prevalence of certain crimes in the year prior, such as robbery with a firearm, assault, and even murder itself, as these crimes would likely create an atmosphere of fear that would sway voters and politicians to pass such a law. In our analysis, we lagged all the crime variables (assault, burglary, motor vehicle theft, murder, robbery, and robbery with a firearm) by one year, so we were able to accurately capture their causal effect on the next year.

Our full DAG illustrating the causal relationships we believe are in effect is presented **Figure 1**.

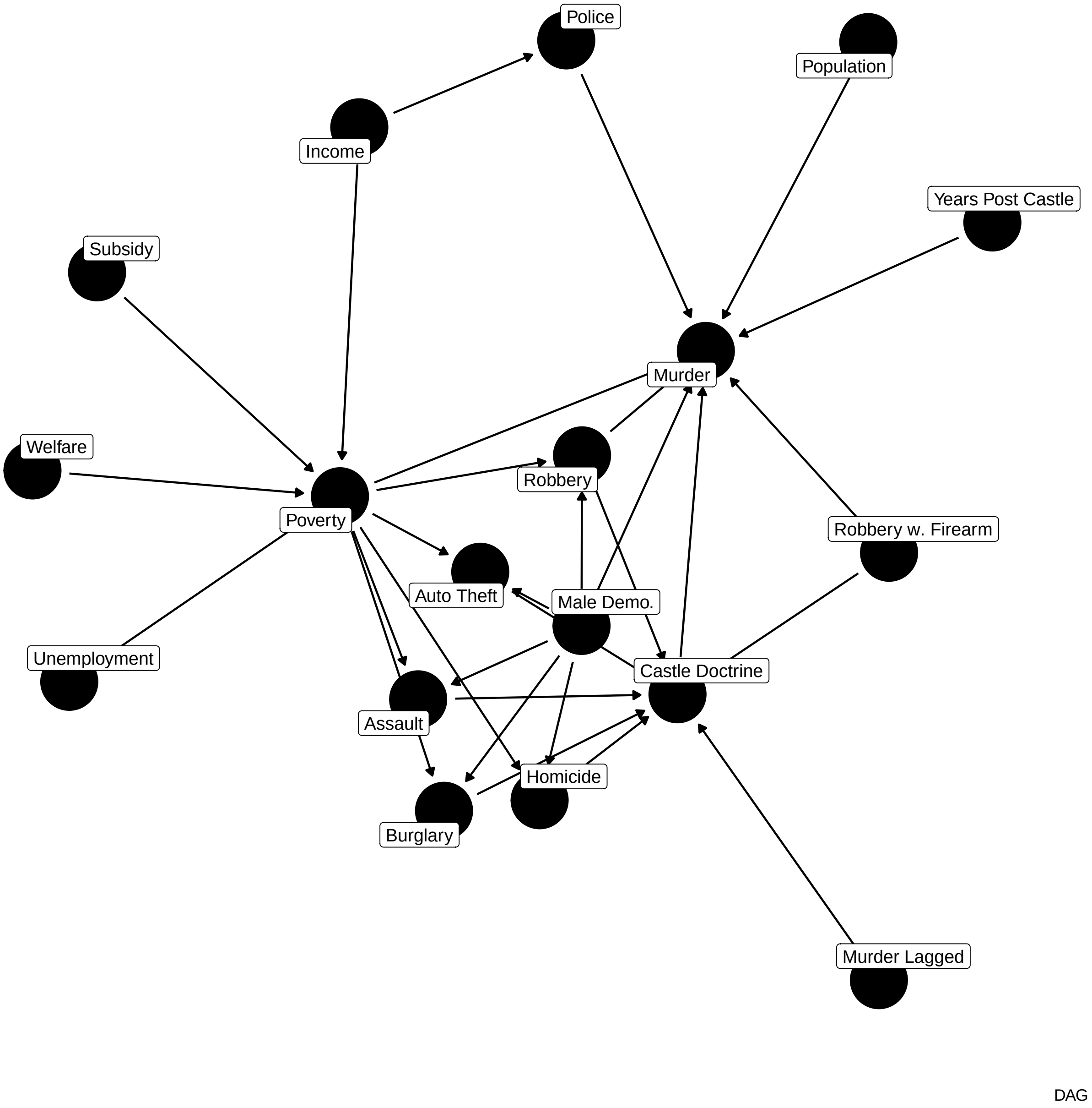

Police
Population
Income
Years Post Castle
Subsidy
Murder
Welfare
Robbery
Poverty
Robbery w. Firearm
Auto Theft
Male Demo.
Unemployment
Castle Doctrine
Assault
Homicide
Burglary
Murder Lagged
DAG

Figure 1: DAG

# Adjustments and Missing Data

## Adjustment Sets

After mapping out our DAG, the next step in our Causal Analysis was identifying potential confounders that we would need to adjust for in our model. There are two possible adjustment sets for our DAG as shown in **Figure 2**. The minimal adjustment set includes homicide, assault, burglary, robbery, robbery involving a firearm, and motor vehicle theft, while the second is made up of robbery, poverty, robbery involving a firearm, and all four male age-race variables (% of Black males age 15-24, % of white males age 15-24, % of black males age 25-44, % of white males age 25-44).

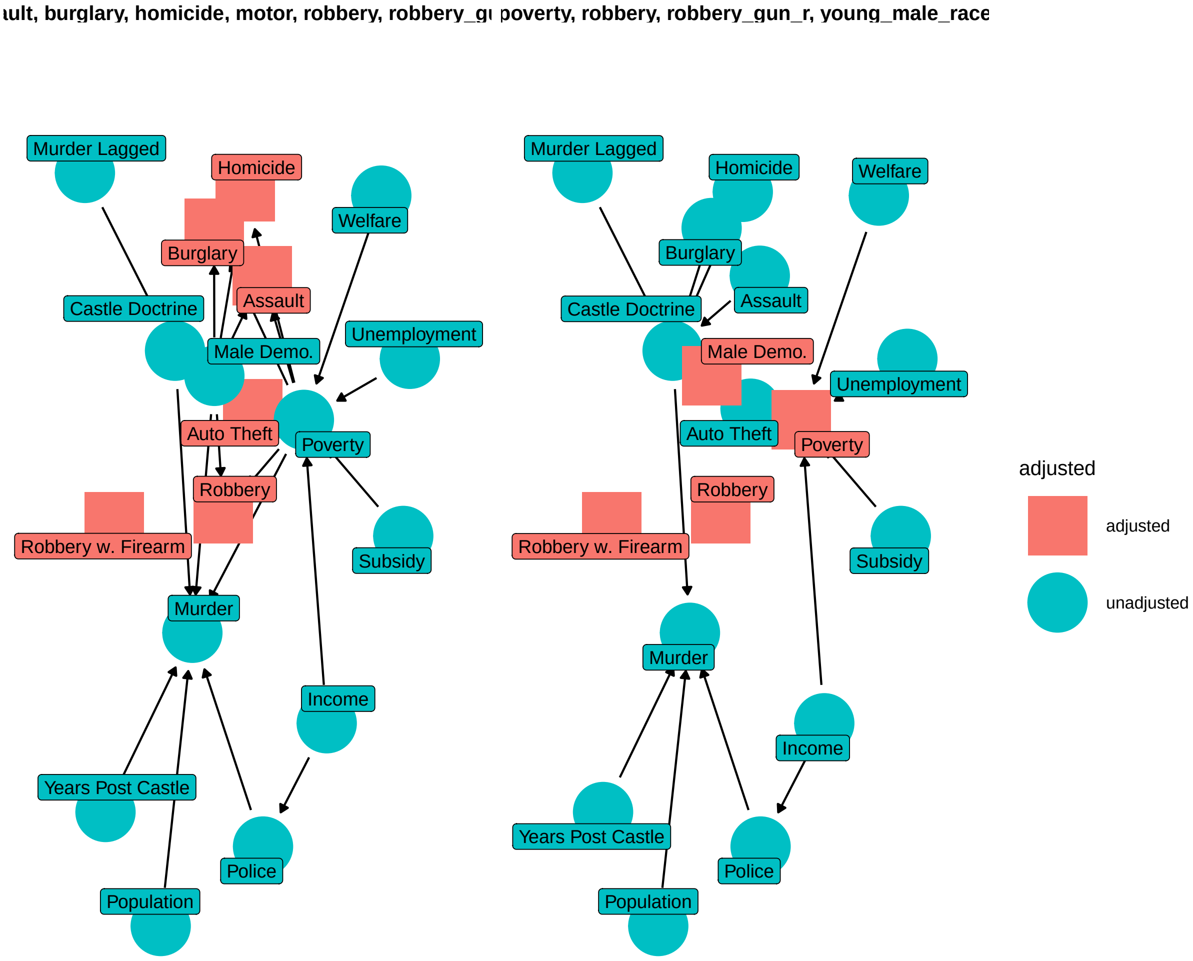


Figure 2: DAG w. Adjustment Sets

## Missing Data

In our analysis of missing data, we found that robbery involving a firearm has about 1.09% missing data (6 observations), and is only missing for one state, Washington. No other variables include missing data. Due to the low level of missingness and the fact that Washington did not implement Castle Doctrine, which is the much larger subpopulation, we proceeded with a complete case analysis and excluded the state of Washington from consideration.

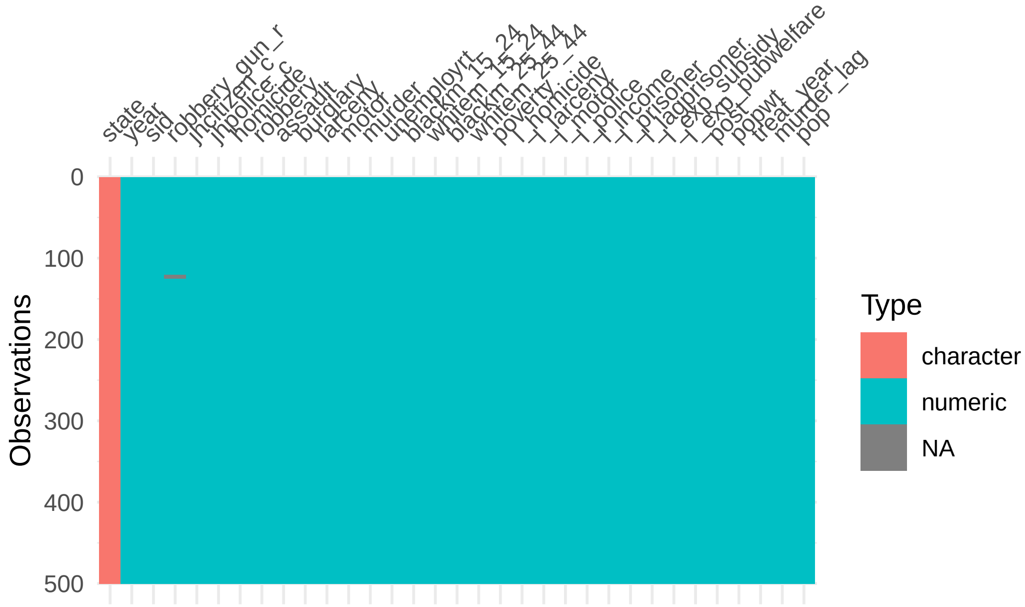


Figure 3: Missing Data Visualization

# Propensity Weighting

We used inverse propensity score weighting (IPW) in our analysis, which allowed us simulate what the relationship between exposure (implementing Castle Doctrine) and outcome (murder rates) would have looked like if our data had come from a randomized trial instead of being observational. Since we targeted the effect of the policy on the treated states, we employed the inverse probability weight for the Average Treatment Effect Among the Treated (ATT) to estimate the effect. Here, the propensity score for each observation is the probability of the state implementing Castle Doctrine given their particular covariate values. To simplify our model and minimize potential measurement error, we used the minimal adjustment set to identify confounders. We then fit a logistic regression to calculate the probabilities of states passing Castle Doctrine with the model shown below:

$$log\_odds_i = S_{1:3}(Homicide_i) + S_{1:2}(Burglary_i) + Assault_i + MotorTheft_i + Robbery_i + ArmedRobbery_i$$

## Unweighted Data

Table 1: **Sample Characteristics by Castle Doctrine**

| **Characteristic** | **Post-Doctrine**<br>N = 74 | **Pre-Doctrine**<br>N = 411 | **Overall**<br>N = 485 |
|---|---|---|---|
| Assault | 312 (188, 456) | 226 (161, 348) | 235 (163, 357) |
| Burglary | 906 (703, 1,008) | 634 (510, 882) | 663 (515, 936) |
| Homicide | 6.14 (4.65, 7.02) | 4.44 (2.46, 6.33) | 4.73 (2.61, 6.50) |
| Motor Vehicle Theft | 310 (225, 378) | 308 (209, 426) | 309 (214, 417) |
| Robbery | 116 (90, 145) | 97 (63, 153) | 100 (67, 152) |
| Robbery w. Firearm | 0.46 (0.37, 0.51) | 0.34 (0.25, 0.45) | 0.36 (0.26, 0.46) |

[1] Median (Q1, Q3); Rates per 100,000 persons

Before weighting our subpopulations with IPW, we observed substantial imbalances between the pre and post-Doctrine subpopulations. From **Table 1**, we saw that the Pre-Doctrine group had 411 observations, with lower median crime rates across all covariates in the adjustment set. This imbalance was also reflected in the distributions of propensity scores, displayed in **Figure 4**. The distribution of propensity scores for states that had not passed Castle Doctrine laws was highly right skewed, with a majority of the scores falling between 0.0 and 0.2, while the distribution for post-Castle Doctrine states was spread relatively uniformly from 0.0 to 0.7.

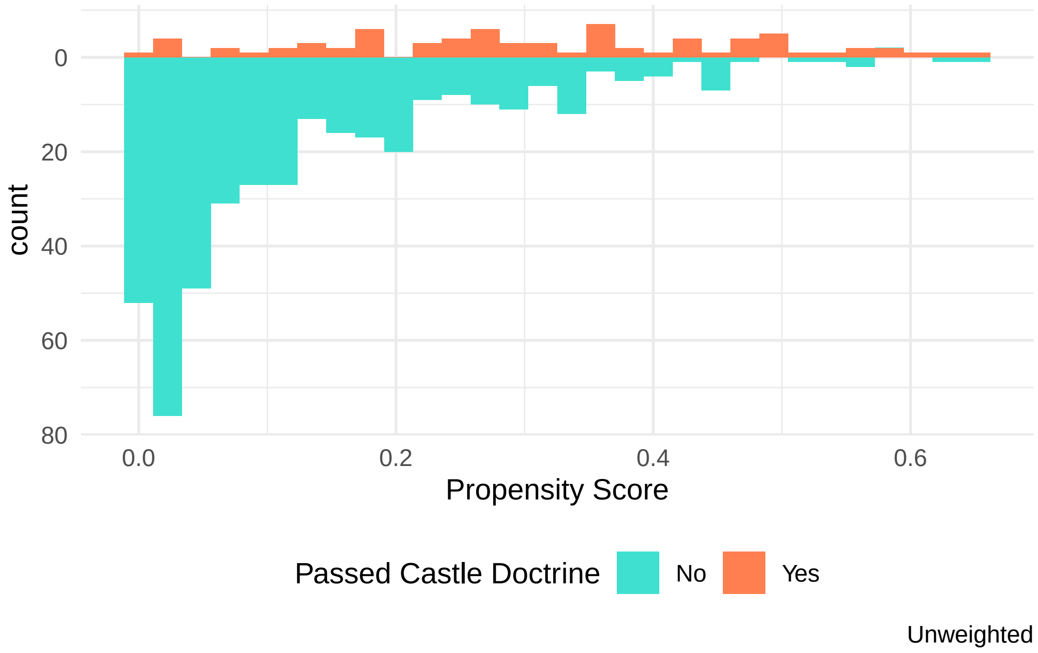


Figure 4: Mirrored Histogram, Unweighted

## ATT Weighting

Table 2: **Sample Characteristics by Re-Weighted Castle Doctrine**

| **Characteristic** | **Post-Doctrine**<br>N = 74; ESS = 74.0 | **Pre-Doctrine**<br>N = 411; ESS = 143.7 | **Overall**<br>N = 485; ESS = 193.3 |
|---|---|---|---|
| Assault | 311 (188, 456) | 310 (207, 404) | 311 (200, 418) |
| Burglary | 903 (703, 1,008) | 825 (639, 1,035) | 881 (661, 1,016) |
| Homicide | 6.07 (4.65, 7.02) | 6.19 (4.79, 7.34) | 6.19 (4.75, 7.16) |
| Motor Vehicle Theft | 309 (225, 378) | 296 (234, 394) | 306 (234, 387) |
| Robbery | 116 (90, 145) | 101 (83, 139) | 109 (85, 144) |
| Armed Robbery | 0.46 (0.37, 0.51) | 0.44 (0.36, 0.49) | 0.45 (0.36, 0.50) |

Abbreviation: ESS = Effective Sample Size

[1] Median (Q1, Q3); Rates per 100,000 persons

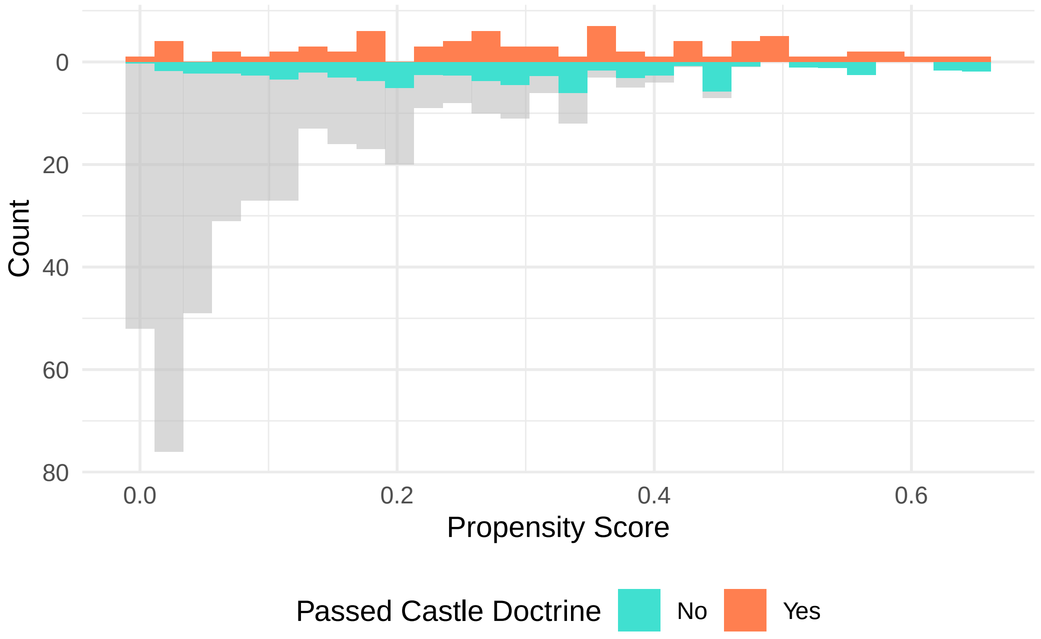


Figure 5: Mirrored Histogram, Unweighted and Weighted

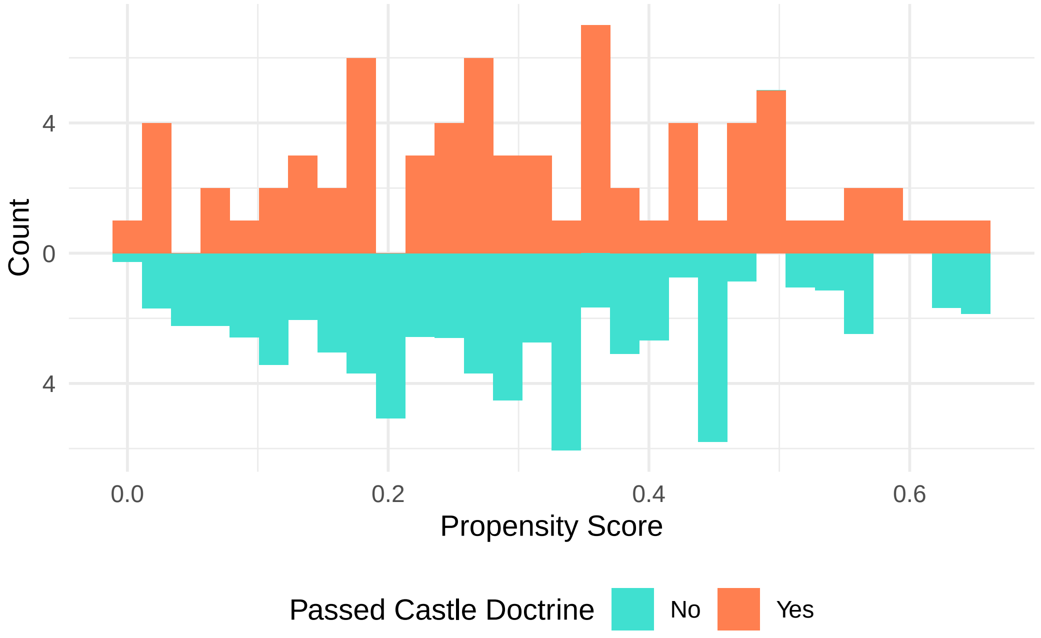


Figure 6: Mirrored Histogram, ATT Only

To account for the differences across the two subpopulations, we fit the previously-specified propensity score model, predicting a state's probability of passing a Castle

Doctrine law based on the covariates in our adjustment set. After reweighting our data, we observed in **Table 2** that the median value of the covariates are much more matched between Castle Doctrine states and non-Castle Doctrine states.

We display the distributions of the propensity scores for the subpopulations in the mirrored histograms in **Figures 5** and **6**. The distributions of propensity scores in the mirrored histograms demonstrated substantially improved balance between the two groups, and we did not observe any alarming positivity issues.

We also used a Love Plot (**Figure 7**) to visualize the effect of ATT weighting on the Standardized Mean Difference for each confounding covariate. Before weighting, we observed the SMD for each covariate except for Robbery was relatively large, reflecting the unbalance between Castle Doctrine states and non-Castle Doctrine states across these variables. After re-weighting, we saw that the SMDs across the covariates are below the 0.1 threshold, suggesting that the weighting method had balanced the two pseudo-populations across each covariate very effectively. These results aligned with the conclusions we were able to draw from comparing the unweighted and re-weighted tables.

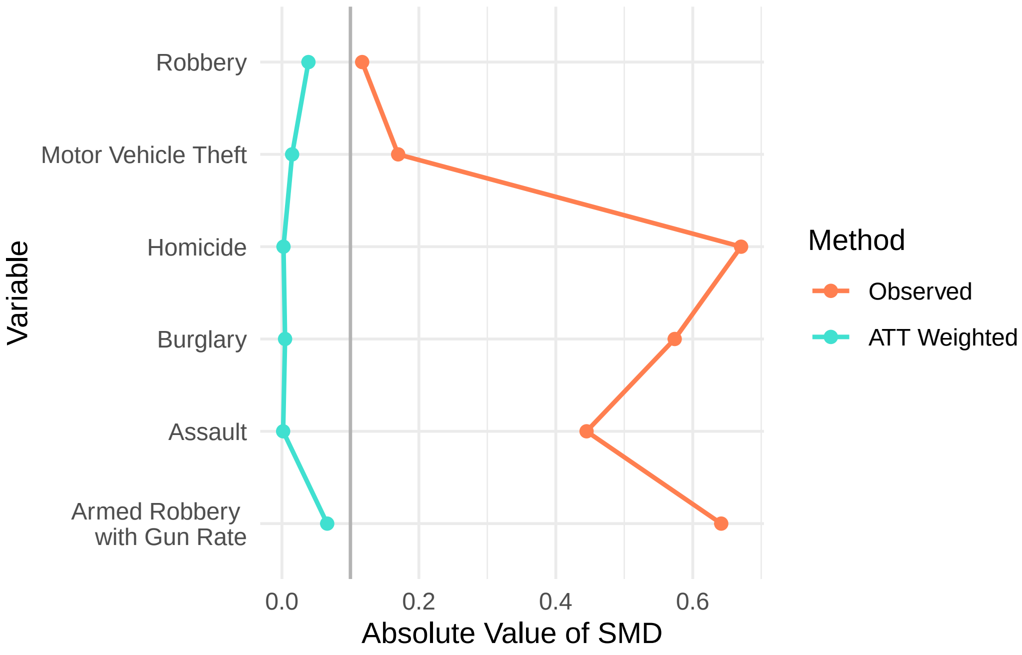


Figure 7: Love Plot

Finally, we compared the empirical cumulative distribution functions (eCDFs) of our confounders. Since all confounders in the adjustment set were continuous, we constructed eCDF plots for each one. Although we were not able to achieve perfect correspondence between the two subpopulations through weighting alone, incorporating second and third degree splines (as reflected in our propensity score model) on Homicide and Burglary helped to bring the eCDFs for those covariates closer together (see Appendix). Overall, our checks suggested that the necessary assumptions for causal inference appeared to be reasonably met, so we proceeded with effect estimation.

## G-Computation

To capture the time-varying effect of the implementation of Castle Doctrine, we performed G-Computation using a linear model that incorporated ATT weights from our IPW model, along with an interaction term representing the number of years before and after enactment of a state's Castle Doctrine law:

$$Murder_i = CastleDoctrine_i + Years_i + CastleDoctrine_i : Years_i$$

We simulated counterfactual data for each of the four years before and after the enactment of Castle Doctrine laws. For each year, we computed the difference in murder rates between the post- and pre-enactment periods for both Castle Doctrine and non–Castle Doctrine states. Using these differences, we conducted a pseudo–difference-in-differences analysis to estimate the treatment effect of Castle Doctrine over time. Based on our model, we estimated the average treatment effect among the treated (ATT) for four time intervals surrounding the law's passage. Four years was the largest symmetric time-span around Castle Doctrine passage available for comparison in our data. The point estimates of these effects are presented in **Table 3**. Across all periods, we observed a negative effect on the murder rate; however, to assess the reliability of these estimates, we conducted an uncertainty analysis

Table 3: Estimated ATT by Year Relative to Castle Doctrine Ratification

| Years Before/<br>After Castle Doctrine | ATT Point Est. |
|---:|---:|
| 1 | -0.797 |
| 2 | -1.593 |
| 3 | -2.390 |
| 4 | -3.186 |

## Measures of Uncertainty

To assess the uncertainty of our estimates, we employed a bootstrap approach, resampling the data 1,000 times to generate a distribution of estimated treatment effects. From this empirical distribution, we calculated the 2.5 and 97.5 percentiles to construct 95% confidence intervals for each point estimate. All four confidence intervals contained zero, indicating that none of the estimated effects were statistically significant at the 95% level.

Table 4: Estimated ATT by Year Relative to Castle Doctrine Ratification

| Years Before/<br>After Castle Doctrine | ATT Estimate | ATT Standard Dev. | 95% CI Lower Bound | Upper Bound |
|---:|---:|---:|---:|---:|
| 1 | -0.781 | 0.584 | -1.881 | 0.342 |
| 2 | -1.546 | 1.137 | -3.847 | 0.535 |
| 3 | -2.323 | 1.737 | -5.580 | 1.180 |
| 4 | -3.110 | 2.370 | -7.829 | 1.510 |

# Sensitivity Analysis:

## Alternate Adjustment Set:

To ensure the robustness of our results, we conducted a sensitivity analysis to assess the validity of our DAG structure and the potential impact of unmeasured confounding. First, we explored the causal effect using the alternate adjustment set from the DAG. If the covariates in the models were well-measured and the DAG was specified correctly, then each adjustment set should yield similar treatment effects, as both should provide an unbiased estimate of the true causal effect.

Using the alternative adjustment set–which included robbery, poverty, robbery involving a firearm, and all four male age-race variables–we refit the propensity score model with a five-degree spline on the poverty variable and a four-degree spline on robbery involving a firearm to help balance the subpopulations. Following the same procedure as in our original analysis, we used G-Computation, difference-in-differences, and bootstrapping to obtain confidence intervals for the average treatment effect among the treated population. In **Table 5**, we display the point estimates from the same four time periods of our original analysis. Compared to the point estimates from the minimal adjustment set in **Table 3**, the effects using the alternate adjustment set are attenuated in magnitude. Nevertheless, examining the 95% confidence intervals in **Table 6**, all confidence intervals still include zero, reinforcing the conclusion that enacting Castle Doctrine did not have a statistically significant impact on murder rate in states that passed such laws.

The discrepancy in point estimates suggested that there may have been measurement error in one or more covariates in the dataset or that the DAG did not fully capture the true underlying causal structure at work. To investigate the latter, we made some modifications to the DAG, such as connecting the police presence variable to crime variables; however, we found these changes did not significantly affect our results.

Table 5: Alt. Estimated ATT by Year Relative to Castle Doctrine Ratification

| Years Before/<br>After Castle Doctrine | ATT Point Est. |
|---:|---:|
| 1 | -0.394 |
| 2 | -0.788 |
| 3 | -1.181 |
| 4 | -1.575 |

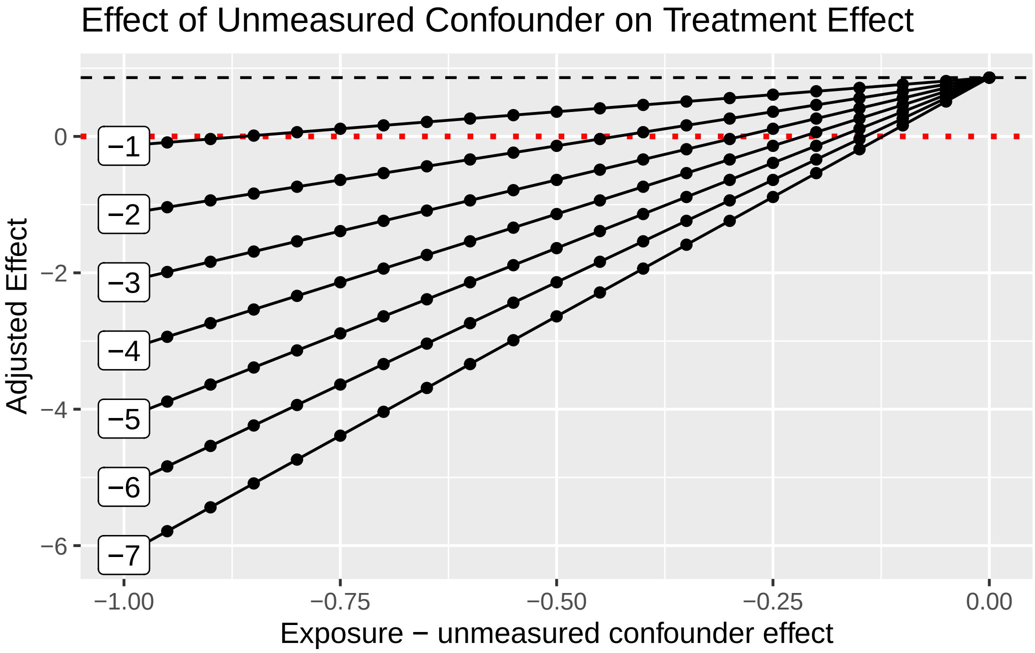
Table 6: Alt. Estimated ATT by Year Relative to Castle Doctrine Ratification

| Years Before/<br>After Castle Doctrine | ATT Estimate | ATT Standard Dev. | 95% CI Lower Bound | Upper Bound |
|---:|---:|---:|---:|---:|
| 1 | -0.399 | 0.618 | -1.559 | 0.862 |
| 2 | -0.810 | 1.190 | -3.101 | 1.462 |
| 3 | -1.193 | 1.769 | -4.362 | 2.490 |
| 4 | -1.616 | 2.385 | -6.166 | 2.761 |

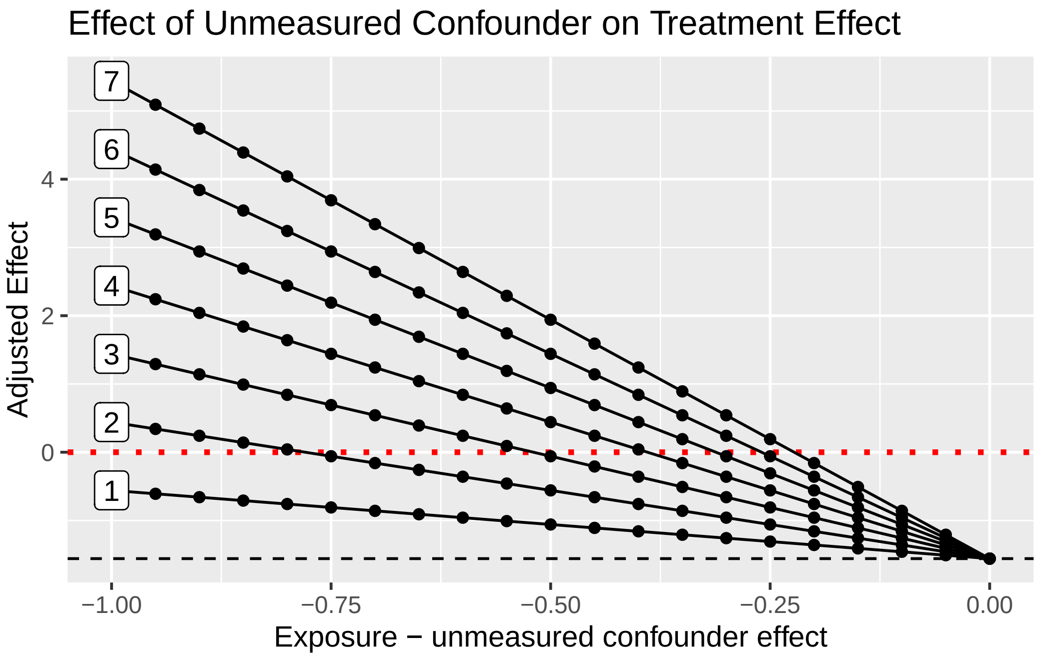
## Tipping Point Analysis

Finally, we assessed the potential impact of an unmeasured confounder on the relationship between Castle Doctrine laws and murder rates using a tipping point analysis. Because the observed effects at 1 through 4 years from implementation were not significant at a 95% significance level, we focused our analysis on the confounding required to to make the estimated effect statistically significant in either a positive or negative direction. For conciseness, we limited the analysis the effect at one year before and after implementation.

The plots below illustrate how varying strength of an unmeasured confounder's associations with both the exposure (Castle Doctrine) and the outcome (Murder Rate) would bias the estimated effect at one year before and after exposure. The graphs allowed us to identify tipping points, where the adjusted treatment effect crossed the null.



Looking at the negative plot (shown above), we observed that, for example, if a one standard deviation change in the unmeasured confounder reduced murder rate by 2 percent and was associated with a probability of 0.45 of not passing a Castle Doctrine law (-0.45 on the x-axis), then the upper bound of the adjusted effect would cross zero, such that the 95% CI for the ATT would be entirely below zero, representing a significant negative effect. Since our dataset was somewhat limited in its scope of variables, we found it reasonable to believe that such a confounder might exist.



Looking at the positive plot (shown above), we observed that, if a one standard deviation change in the unmeasured confounder increased murder rate by 3 percent and was associated with a probability of 0.55 of not passing a Castle Doctrine law (-0.55 on the x-axis), then the lower bound of the adjusted effect would cross zero, such that the 95% CI for the ATT would be entirely positive, representing a significant positive effect on the murder rate. Such a confounder is somewhat less likely than the previous; however, again, given the limitations in our data, more investigation would be needed to completely rule one out.

# Conclusions

We find the implementation of Castle Doctrine did not have a statistically significant effect on murder rates in U.S. states that enacted such laws between 2001 and 2010. Across four distinct intervals of years before and after enactment, we observed no significant effects on murder rates.

However, several limitations may affect the validity of our findings. First, our DAG reflects our assumptions about the underlying causal structure in question. Consulting subject matter experts to help validate and refine relationships among key variables would be needed to improve further analysis. Similarly, our analysis was constrained by data availability. Important covariates—such as state-level gun ownership rates and political leaning—were not included but were likely relevant to the causal dynamics at play. Additionally, while all the laws in our dataset fall under the umbrella of “Castle Doctrine,” they vary in their specific provisions, which may influence the effects of such laws on violent crime within their respective states. In fact, the documentation of the dataset refers to the variable in question as the implementation of Castle Doctrine statutes, whereas Cheng and Hoekstra appear to group Castle Doctrine and Stand Your Ground laws together (Huntington-Klein and Barrett 2024; Cheng and Hoekstra 2013). This distinction may be important, as the two types of laws, while similar, have meaningful differences that could be obscured by this grouping. For future analyses, we recommend compiling a more comprehensive dataset to support more robust and granular modeling.

Finally, our focus on murder rates alone presents another limitation. As of 2017, only about 100 burglary-homicides occurred annually nationwide, suggesting our analysis may be underpowered to detect an effect of Castle Doctrine on murder (Gillin 2017). Examining other outcomes related to Castle Doctrine—such as burglary rates or overall homicide rates—could provide a more comprehensive understanding of the laws’ impact on violent crime.

## References

Cheng, C., and M. Hoekstra. 2013. “Does Strengthening Self-Defense Law Deter Crime or Escalate Violence? Evidence from Expansions to Castle Doctrine.” *The Journal of Human Resources* 48 (3): 821–53. http://www.jstor.org/stable/23799103.

Gillin, J. 2017. “Moore Flubs Stats on Gun Deaths During Home Invasions.” October 25, 2017. https://www.politifact.com/factchecks/2017/oct/25/michael-moore/michael-moore-flubs-stats-people-killed-guns-durin/.

Huntington-Klein, N., and M. Barrett. 2024. “Castle Dataset. R PACKAGES.” October 24, 2024. https://r-packages.io/datasets/castle.

Jay, R. 2006. “Spitting Lead in Leadville: Doc Holliday’s Last Stand.” August 14, 2006. https://www.historynet.com/spitting-lead-in-leadville-doc-hollidays-last-stand/?f.

Johnson, D. 1990. “Colorado Journal; ”Make My Day”: More Than a Threat.” June 1, 1990. https://www.nytimes.com/1990/06/01/us/colorado-journal-make-my-day-more-than-a-threat.html.

Shouse, N. 2025. “Castle Doctrine Vs. Stand Your Ground—What’s the Difference? Shouse Law Group - Criminal, Immigration, Injury & Employment Lawyers.” February 4, 2025. https://www.shouselaw.com/ca/blog/castle-doctrine-vs-stand-your-ground/.

# Appendix

```r
# Libraries

library(causaldata)
library(dagitty)
library(ggdag)
library(tidyverse)
library(knitr)
library(naniar)
library(gtsummary)
library(gt)
library(broom)
library(readxl)
library(propensity)
library(halfmoon)
library(patchwork)
library(visdat)
library(survey)
library(labelled)
library(tipr)
library(kableExtra)

data("castle")

# Data transformations

castle$treat_year <- ifelse(castle$post == 1, castle$year, 0)

lower <- 1
upper <- 11
i <- 1
while(i < 51){
  treat_year_1 <-
min(castle$treat_year[lower:upper][castle$treat_year[lower:upper] !=0])
  castle$treat_year[lower:upper] <- rep(treat_year_1, 11)
  lower <- upper + 1
  upper <- lower + 10
  i <- i + 1
```

```r
}

castle <- castle |>
  mutate(years_after_treat = year - treat_year)

castle$years_after_treat <-
  ifelse(castle$years_after_treat == -Inf, NA, castle$years_after_treat)


castle <- castle |>
  group_by(sid) |>
  mutate_at(c("assault", "burglary",
              "homicide", "larceny",
              "motor", "l_larceny",
              "l_motor", "l_police",
              "l_income", "l_exp_subsidy",
              "l_exp_pubwelfare"), lag) |>
  mutate(murder_lag = lag(murder)) |>
  ungroup()


castle$sid <- ifelse(as.numeric(castle$sid) > 8,
                     castle$sid - 1, castle$sid)

state_id_list <- read_excel("state_id_list_fixed.xlsx",
                            col_names = FALSE)

colnames(state_id_list) <- c("state", "pop", "sid")

state_id_ranks <- state_id_list |>
  select(state, sid)
castle_dat <- full_join(castle, state_id_list, by = "sid")

castle_dat <- castle_dat[castle_dat$year != 2000,]

castle_for_tab <- castle_dat |>
  select(!(starts_with("r20") |
           starts_with("trend") |
           starts_with("lead") |
           starts_with("lag") |
             starts_with("years")))

#CCA + Dropping Washington
castle_dat <- castle_dat |>
  select(!(starts_with("r20") |
             starts_with("trend") |
             starts_with("lead") |
             starts_with("lag"))) |>
  drop_na(robbery_gun_r) |>
  filter(sid != 47)


#DAG structure

# young_male_race is taking place of:
# blackm_15_24 + whitem_15_24 + blackm_25_44 + whitem_25_44

castle_dag <- dagify(
  murder ~ young_male_race + poverty + popwt +
          robbery_gun_r + l_police + post +
          years_after_treat + robbery,

  post ~ homicide + robbery_gun_r + assault +
          burglary + motor + murder_lag + robbery,

  burglary ~ poverty + young_male_race,
  homicide ~ poverty + young_male_race,
  motor ~ poverty + young_male_race,
  robbery ~ poverty + young_male_race,
  assault ~ poverty + young_male_race,

  poverty ~ unemployrt + l_exp_subsidy +
            l_exp_pubwelfare + l_income,

  l_police ~ l_income,

  outcome = "murder",
  exposure = "post",
  labels = c(
    murder = "Murder",
    murder_lag = "Murder Lagged",
    unemployrt = "Unemployment",
    young_male_race = "Male Demo.",
    poverty = "Poverty",
    popwt = "Population",
    robbery_gun_r = "Robbery w. Firearm",
    l_exp_subsidy = "Subsidy",
    l_exp_pubwelfare = "Welfare",
    l_police = "Police",
    post = "Castle Doctrine",
    years_after_treat = "Years Post Castle",

    homicide = "Homicide",
    robbery = "Robbery",
    assault = "Assault",
    burglary = "Burglary",
    motor = "Auto Theft",
    l_income = "Income")
)


#DAG

ggdag(castle_dag, layout = "nicely",
      use_labels = "label", text = FALSE) +
  labs(caption = "DAG") +
  theme_dag()


# Adjustment Sets

ggdag_adjustment_set(castle_dag, text_col = "black",
                     use_labels = "label",
                     text = FALSE) +
  theme_dag()


# Missing Data Viz

vis_dat(castle_for_tab)
```

```
# Unweighted Table

castle_select <- castle_dat |>
  select(c(post, assault, burglary, homicide,
           motor, robbery, robbery_gun_r)) |>
  mutate(post = ifelse(post == 0,
                       "Pre-Doctrine",
                       "Post-Doctrine")) |>
  set_variable_labels(
    post = "Passage of Castle Doctrine",
    assault = "Assault",
    burglary = "Burglary",
    homicide = "Homicide",
    motor = "Motor Vehicle Theft",
    robbery = "Robbery",
    robbery_gun_r = "Robbery w. Firearm"
  )

tbl_summary(
  castle_select,
  by = post
) |>
  add_overall(last = TRUE) |>
  modify_caption("**Sample Characteristics by Castle Doctrine**") |>
  modify_footnote(everything() ~
                    "Median (Q1, Q3); Rates per 100,000 persons") |>
  as_kable_extra(format = "latex") |>
  kable_styling(
    latex_options = c("scale_down", "hold_position"),
    full_width = FALSE,
    position = "center"
  )


# Propensity model

propensity_model <- glm(post ~ splines::ns(homicide, 3) +
                          splines::ns(burglary, 2) +
                          assault + motor + robbery +
                          robbery_gun_r ,
                        data = castle_dat,
                        family = "binomial")

castle_dat <- propensity_model |>
  augment(type.predict = "response", data = castle_dat) |>
  mutate(w_att = wt_att(.fitted, post, exposure_type = "binary"))


# Mirrored histogram, unweighted

ggplot(castle_dat, aes(x = .fitted,
                       group = post,
                       fill = post)) +
  geom_mirror_histogram(bins = 30,
                        alpha = 1,
                        aes(fill = factor(post))) +
  labs(x = "Propensity Score",
       fill = "Passed Castle Doctrine",
       caption = "Unweighted") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                    values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")


# Weighted table

castle_select2 <- castle_dat |>
  select(c(post, assault, burglary,
           homicide, motor, robbery,
           robbery_gun_r, w_att)) |>
  mutate(post = ifelse(post == 0,
                       "Pre-Doctrine",
                       "Post-Doctrine")) |>
    set_variable_labels(
    post = "Passage of Castle Doctrine",
    assault = "Assault",
    burglary = "Burglary",
    homicide = "Homicide",
    motor = "Motor Vehicle Theft",
    robbery = "Robbery",
    robbery_gun_r = "Armed Robbery"
  )

svy_des <- svydesign(
  ids = ~1,
  data = castle_select2,
  weights = ~w_att
)

hdr <- paste0(
  "**{level}**  \n",
  "N = {n_unweighted}; ESS = {format(n, digits = 1, nsmall = 1)}"
)

tbl_svysummary(svy_des,
               by = post,
               include = c(assault, burglary,
                           homicide, motor,
                           robbery, robbery_gun_r))|>
    add_overall(last = TRUE) |>
  add_ess_header(header = hdr) |>
modify_caption("**Sample Characteristics by Re-Weighted Castle Doctrine**") |>
  modify_footnote(everything() ~
                    "Median (Q1, Q3); Rates per 100,000 persons")|>
  as_kable_extra(format = "latex") |>
  kable_styling(
  latex_options = c("scale_down", "hold_position"),
  full_width = FALSE,
  position = "center"
  )


# Mirrored Hist Weighted

ggplot(castle_dat, aes(x = .fitted,
                       group = post,
                       fill = post)) +
  geom_mirror_histogram(bins = 30,
                        alpha = .6,
                        fill = "grey") +
  labs(x = "Propensity Score",
       y = "Count") +
  geom_mirror_histogram(bins = 30,
                        alpha = 1,
                        aes(fill = factor(post),
                            weight = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Propensity Score",
       y = "Count",
       fill = "Passed Castle Doctrine") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                    values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")


#Mirrored Histogram, ATT Only

ggplot(castle_dat, aes(x = .fitted,
                       group = post,
                       fill = post)) +
  geom_mirror_histogram(bins = 30,
                        alpha = 1,
                        aes(fill = factor(post),
                            weight = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Propensity Score",
       y = "Count",
       fill = "Passed Castle Doctrine") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                    values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
#Organizing Love Plot

dat_for_love <- castle_dat |>
  select(assault, burglary,
         homicide, motor,
         robbery, robbery_gun_r,
         post, w_att)

colnames(dat_for_love) <- c("Assault", "Burglary",
                            "Homicide", "Motor Vehicle Theft",
                            "Robbery", "Armed Robbery \n with Gun Rate",
                            "post", "w_att")

weighted_for_love <- tidy_smd(
  dat_for_love,
  .vars = c(Assault, Burglary,
            Homicide, `Motor Vehicle Theft`,
            Robbery, `Armed Robbery \n with Gun Rate`),
  .group = post,
  .wts = c(w_att)
)

#Love Plot

ggplot(data = weighted_for_love,
       aes(x = abs(smd),
           y = variable,
           group = method,
           color = method)) +
  geom_love() +
  scale_color_manual(values = c("coral", "turquoise"),
                     labels = c("Observed", "ATT Weighted")) +
  labs(color = "Method",
       x = "Absolute Value of SMD",
       y = "Variable") +
  theme_minimal()


#G-Computation
```

```
standardized_model <- lm(murder ~ post*years_after_treat,
                         data = castle_dat,
                         weights = w_att)

years_pre_post <- c(1:4)

estimate_df <- data.frame("years_pre_post" = years_pre_post,
                           "ATT_est" = rep(NA, 4))

for (j in years_pre_post){

    standardized_model <- lm(murder ~ post*years_after_treat,
                        data = castle_dat, weights = w_att)

    castle_dat_minus_j_yes <- castle_dat |>
      mutate(years_after_treat = -1*j, post = 1)

    castle_dat_minus_j_no <- castle_dat |>
      mutate(years_after_treat = -1*j, post = 0)

    castle_dat_plus_j_yes <- castle_dat |>
      mutate(years_after_treat = j, post = 1)

    castle_dat_plus_j_no <- castle_dat |>
      mutate(years_after_treat = j, post = 0)

    new_data_plus_j_yes <- standardized_model |>
      augment(newdata = castle_dat_plus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_plus_j_no <- standardized_model |>
      augment(newdata = castle_dat_plus_j_no) |>
      rename(murder_est = .fitted)

    new_data_minus_j_yes <- standardized_model |>
      augment(newdata = castle_dat_minus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_minus_j_no <- standardized_model |>
      augment(newdata = castle_dat_minus_j_no) |>
      rename(murder_est = .fitted)

    estimate_df$years_pre_post[j] <- j

    estimate_df$ATT_est[j] <-
      (mean(new_data_plus_j_yes$murder_est) -
       mean(new_data_minus_j_yes$murder_est)) -
    (mean(new_data_plus_j_no$murder_est) -
       mean(new_data_minus_j_no$murder_est))
}

estimate_df$ATT_est <- round(estimate_df$ATT_est, 3)

colnames(estimate_df) <-
  c("Years Before/\\\\After Castle Doctrine",
                           "ATT Point Est.")


estimate_df |>
  kbl(format = "latex",
      booktabs = TRUE,
      escape = F,
caption = "Estimated ATT by Year Relative to Castle Doctrine Ratification") |>
  kable_styling(
    latex_options = c("scale_down", "hold_position"),
    full_width = FALSE,
    position = "center"
  )


#Bootstrap CIs

set.seed(779)

n_bootstrap <- 1000

bootstrap_df <- data.frame("years_pre_post" = years_pre_post,
                           "mean_ATT" = rep(NA, 4),
                           "sd_ATT" = rep(NA, 4),
                           "ci.l" = rep(NA, 4),
                           "ci.u" = rep(NA, 4))

for (j in years_pre_post){

  bootstrap_est <- vector(length = n_bootstrap)

  for (b in 1:n_bootstrap){
    boot_dat <- castle_dat |>
      slice_sample(n = nrow(castle_dat), replace = T)

    standardized_boot <- lm(murder ~ post*years_after_treat,
                        data = boot_dat, weights = w_att)

    boot_dat_minus_j_yes <- boot_dat |>
      mutate(years_after_treat = -1*j, post = 1)

    boot_dat_minus_j_no <- boot_dat |>
      mutate(years_after_treat = -1*j, post = 0)

    boot_dat_plus_j_yes <- boot_dat |>
      mutate(years_after_treat = j, post = 1)

    boot_dat_plus_j_no <- boot_dat |>
      mutate(years_after_treat = j, post = 0)

    new_data_plus_j_yes <- standardized_boot |>
      augment(newdata = boot_dat_plus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_plus_j_no <- standardized_boot |>
      augment(newdata = boot_dat_plus_j_no) |>
      rename(murder_est = .fitted)

    new_data_minus_j_yes <- standardized_boot |>
      augment(newdata = boot_dat_minus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_minus_j_no <- standardized_boot |>
      augment(newdata = boot_dat_minus_j_no) |>
      rename(murder_est = .fitted)
    bootstrap_est[b] <-
      (mean(new_data_plus_j_yes$murder_est) -
        mean(new_data_minus_j_yes$murder_est)) -
      (mean(new_data_plus_j_no$murder_est)
       - mean(new_data_minus_j_no$murder_est))

  }

  bootstrap_df$mean_ATT[j] <- round(mean(bootstrap_est),3)

  bootstrap_df$sd_ATT[j] <- round(sd(bootstrap_est),3)

  bootstrap_df$ci.l[j] <- round(quantile(bootstrap_est, 0.025),3)

  bootstrap_df$ci.u[j] <- round(quantile(bootstrap_est, 0.975),3)
}

colnames(bootstrap_df) <- c("Years Before/\\\\After Castle Doctrine",
                            "ATT Estimate",
                            "ATT Standard Dev.",
                            "95\\% CI Lower Bound",
                            "Upper Bound")

bootstrap_df |>
  kbl(format = "latex",
      booktabs = TRUE,
      escape = FALSE,
caption = "Estimated ATT by Year Relative to Castle Doctrine Ratification") |>
  kable_styling(
    latex_options = c("scale_down", "hold_position"),
    full_width = FALSE,
    position = "center"
  )


#Alt. Adjustment Set Propensity

propensity_model_alt <- glm(post ~ splines::ns(poverty, 5) +
                              robbery + splines::ns(robbery_gun_r, 4) +
                              whitem_15_24 + blackm_15_24 +
                              whitem_25_44 + blackm_25_44,
                            data = castle_dat,
                            family = "binomial")

castle_dat_alt <- propensity_model_alt |>
  augment(type.predict = "response",
          data = castle_dat) |>
  mutate(w_att = wt_att(.fitted,
                        post,
                        exposure_type = "binary"))


# Alt Adjustment Set G-Comp

years_pre_post <- c(1:4)

estimate_df <- data.frame("years_pre_post" = years_pre_post,
                           "ATT_est" = rep(NA, 4))

for (j in years_pre_post){

    standardized_model <- lm(murder ~ post*years_after_treat,
                        data = castle_dat_alt, weights = w_att)

    castle_dat_minus_j_yes <- castle_dat_alt |>
      mutate(years_after_treat = -1*j, post = 1)

    castle_dat_minus_j_no <- castle_dat_alt |>
      mutate(years_after_treat = -1*j, post = 0)

    castle_dat_plus_j_yes <- castle_dat_alt |>
      mutate(years_after_treat = j, post = 1)

    castle_dat_plus_j_no <- castle_dat_alt |>
      mutate(years_after_treat = j, post = 0)

    new_data_plus_j_yes <- standardized_model |>
      augment(newdata = castle_dat_plus_j_yes) |>
      rename(murder_est = .fitted)
    new_data_plus_j_no <- standardized_model |>
      augment(newdata = castle_dat_plus_j_no) |>
      rename(murder_est = .fitted)

    new_data_minus_j_yes <- standardized_model |>
      augment(newdata = castle_dat_minus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_minus_j_no <- standardized_model |>
      augment(newdata = castle_dat_minus_j_no) |>
      rename(murder_est = .fitted)

    estimate_df$years_pre_post[j] <- j

    estimate_df$ATT_est[j] <-
      (mean(new_data_plus_j_yes$murder_est) -
         mean(new_data_minus_j_yes$murder_est)) -
      (mean(new_data_plus_j_no$murder_est) -
         mean(new_data_minus_j_no$murder_est))
}

estimate_df$ATT_est <- round(estimate_df$ATT_est, 3)

colnames(estimate_df) <- c("Years Before/\\\\After Castle Doctrine",
                           "ATT Point Est.")

estimate_df |>
  kbl(format = "latex", booktabs = TRUE, escape = F,
caption = "Alt. Estimated ATT by Year Relative to Castle Doctrine Ratification") |>
  kable_styling(
    latex_options = c("scale_down", "hold_position"),
    full_width = FALSE,
    position = "center"
  )


#Bootstrap Alt.

set.seed(779)
```

```
n_bootstrap <- 1000

bootstrap_df <- data.frame("years_pre_post" = years_pre_post,
                           "mean_ATT" = rep(NA, 4),
                           "sd_ATT" = rep(NA, 4),
                           "ci.l" = rep(NA, 4),
                           "ci.u" = rep(NA, 4))

bootstrap_df_names_clean <- bootstrap_df

for (j in years_pre_post){

  bootstrap_est <- vector(length = n_bootstrap)

  for (b in 1:n_bootstrap){
    boot_dat <- castle_dat_alt |>
      slice_sample(n = nrow(castle_dat_alt), replace = T)

    standardized_boot <- lm(murder ~ post*years_after_treat,
                        data = boot_dat, weights = w_att)

    boot_dat_minus_j_yes <- boot_dat |>
      mutate(years_after_treat = -1*j, post = 1)

    boot_dat_minus_j_no <- boot_dat |>
      mutate(years_after_treat = -1*j, post = 0)

    boot_dat_plus_j_yes <- boot_dat |>
      mutate(years_after_treat = j, post = 1)

    boot_dat_plus_j_no <- boot_dat |>
      mutate(years_after_treat = j, post = 0)

    new_data_plus_j_yes <- standardized_boot |>
      augment(newdata = boot_dat_plus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_plus_j_no <- standardized_boot |>
      augment(newdata = boot_dat_plus_j_no) |>
      rename(murder_est = .fitted)
    new_data_minus_j_yes <- standardized_boot |>
      augment(newdata = boot_dat_minus_j_yes) |>
      rename(murder_est = .fitted)

    new_data_minus_j_no <- standardized_boot |>
      augment(newdata = boot_dat_minus_j_no) |>
      rename(murder_est = .fitted)

    bootstrap_est[b] <-
      (mean(new_data_plus_j_yes$murder_est) -
        mean(new_data_minus_j_yes$murder_est)) -
      (mean(new_data_plus_j_no$murder_est) -
         mean(new_data_minus_j_no$murder_est))

  }

  ## Rounded
  bootstrap_df_names_clean$mean_ATT[j] <-
    round(mean(bootstrap_est),3)

  bootstrap_df_names_clean$sd_ATT[j] <-
    round(sd(bootstrap_est),3)

  bootstrap_df_names_clean$ci.l[j] <-
    round(quantile(bootstrap_est, 0.025),3)

  bootstrap_df_names_clean$ci.u[j] <-
    round(quantile(bootstrap_est, 0.975),3)

  ## Unrounded
  bootstrap_df$mean_ATT[j] <- mean(bootstrap_est)

  bootstrap_df$sd_ATT[j] <- sd(bootstrap_est)

  bootstrap_df$ci.l[j] <- quantile(bootstrap_est, 0.025)

  bootstrap_df$ci.u[j] <- quantile(bootstrap_est, 0.975)
}

colnames(bootstrap_df_names_clean) <-
```

```
  c("Years Before/\\\\After Castle Doctrine",
    "ATT Estimate", "ATT Standard Dev.",
    "95\\% CI Lower Bound", "Upper Bound")

bootstrap_df_names_clean |>
  kbl(format = "latex",
      booktabs = TRUE,
      escape = FALSE,
caption = "Alt. Estimated ATT by Year Relative to Castle Doctrine Ratification") |>
  kable_styling(
    latex_options = c("scale_down", "hold_position"),
    full_width = FALSE,
    position = "center"
  )


#Unmeasured confounder to make significant
#negative effect - 1 year pre/post

ci.u <- bootstrap_df[1,"ci.u"]

adjust_df <- adjust_coef(
  effect_observed = ci.u,
  exposure_confounder_effect = rep(seq(0, -1, by = -0.05), each = 7),
  confounder_outcome_effect = rep(seq(-1, -7, by = -1), times = 21),
  verbose = FALSE
)

ggplot(
  adjust_df,
  aes(
    x = exposure_confounder_effect,
    y = effect_adjusted,
    group = confounder_outcome_effect
  )
) +
  geom_hline(yintercept = ci.u, lty = 2) +
  geom_hline(yintercept = 0, lty = 3, color = "red", lwd = 1) +
  geom_point() +
  geom_line() +
  geom_label(
    data = adjust_df[141:147, ],
    aes(
      x = exposure_confounder_effect,
      y = effect_adjusted,
      label = confounder_outcome_effect
    )
  ) +
  labs(
    x = "Exposure - unmeasured confounder effect",
    y = "Adjusted Effect",
    title = "Effect of Unmeasured Confounder on Treatment Effect"
  )



#Unmeasured confounder to make
#significant positive effect - 1 year pre/post

ci.l <- bootstrap_df[1,"ci.l"]

adjust_df <- adjust_coef(
  effect_observed = ci.l,
  exposure_confounder_effect = rep(seq(0, -1, by = -0.05), each = 7),
  confounder_outcome_effect = rep(seq(1, 7, by = 1), times = 21),
  verbose = FALSE
)

ggplot(
  adjust_df,
  aes(
    x = exposure_confounder_effect,
    y = effect_adjusted,
    group = confounder_outcome_effect
  )
) +
  geom_hline(yintercept = ci.l, lty = 2) +
  geom_hline(yintercept = 0, lty = 3, color = "red", lwd = 1) +
  geom_point() +
  geom_line() +
  geom_label(
    data = adjust_df[141:147, ],
    aes(
      x = exposure_confounder_effect,
      y = effect_adjusted,
      label = confounder_outcome_effect
    )
  ) +
  labs(
    x = "Exposure - unmeasured confounder effect",
    y = "Adjusted Effect",
    title = "Effect of Unmeasured Confounder on Treatment Effect"
  )


#Missing data table

miss_tab <- miss_var_summary(castle_for_tab)
colnames(miss_tab) <- c("Variable", "Missing (n)", "Percent Missing")

miss_tab |>
  gt()


#Before splines

propensity_model_test <- glm(post ~ homicide + burglary +
                assault + motor + robbery + robbery_gun_r,
                data = castle_dat, family = "binomial")

castle_dat_test <- propensity_model_test |>
  augment(type.predict = "response",
          data = castle_dat) |>
  mutate(w_att = wt_att(.fitted,
                        post,
                        exposure_type = "binary"))

ggplot(castle_dat_test, aes(x = .fitted,
                            group = post,
                            fill = post)) +
  geom_mirror_histogram(bins = 30, alpha = .6, fill = "grey") +
  geom_mirror_histogram(bins = 30, alpha = 1,
                        aes(fill = factor(post),
                        weight = w_att)) +
  labs(x = "Propensity Score",
        fill = "Passed Castle Doctrine",
        caption = "ATT", y = "Count") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                    values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")

ggplot(castle_dat_test, aes(x = .fitted,
                            group = post,
                            fill = post)) +
  geom_mirror_histogram(bins = 30,
                        alpha = 1,
                        aes(fill = factor(post),
                        weight = w_att)) +
  labs(x = "Propensity Score",
      fill = "Passed Castle Doctrine",
      caption = "ATT",
      y = "Count") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                    values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")


# Love plot before splines

dat_for_love <- castle_dat_test |>
  select(assault, burglary, homicide,
        motor, robbery, robbery_gun_r,
        post, w_att)

colnames(dat_for_love) <- c("Assault", "Burglary",
                            "Homicide", "Motor Vehicle Theft",
                              "Robbery", "Armed Robbery \n with Gun Rate",
                              "post", "w_att")
weighted_for_love <- tidy_smd(
  castle_dat_test,
  .vars = c(assault, burglary, homicide,
            motor, robbery, robbery_gun_r),
  .group = post,
  .wts = c(w_att)
)

ggplot(data = weighted_for_love,
              aes(x = abs(smd), y = variable,
                  group = method, color = method)) +
  geom_love() +
  theme_minimal()


#eCDFs Pre-splines

p1 <- ggplot(castle_dat_test,
            aes(x = burglary,
            color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Burglary", color = "Castle Implemented") +
  theme_minimal()

p2 <- ggplot(castle_dat_test, aes(x = burglary, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Burglary", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p3 <- ggplot(castle_dat_test, aes(x = assault, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Assault", color = "Castle Implemented") +
  theme_minimal()
```

```r
p4 <- ggplot(castle_dat_test, aes(x = assault, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Assault", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p5 <- ggplot(castle_dat_test, aes(x = robbery, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Robbery", color = "Castle Implemented") +
  theme_minimal()

p6 <- ggplot(castle_dat_test, aes(x = robbery, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Robbery", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p7 <- ggplot(castle_dat_test, aes(x = homicide, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Homicide", color = "Castle Implemented") +
  theme_minimal()

p8 <- ggplot(castle_dat_test, aes(x = homicide, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Homicide", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p9 <- ggplot(castle_dat_test, aes(x = robbery_gun_r,
              color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Robbery with Gun", color = "Castle Implemented") +
  theme_minimal()
```

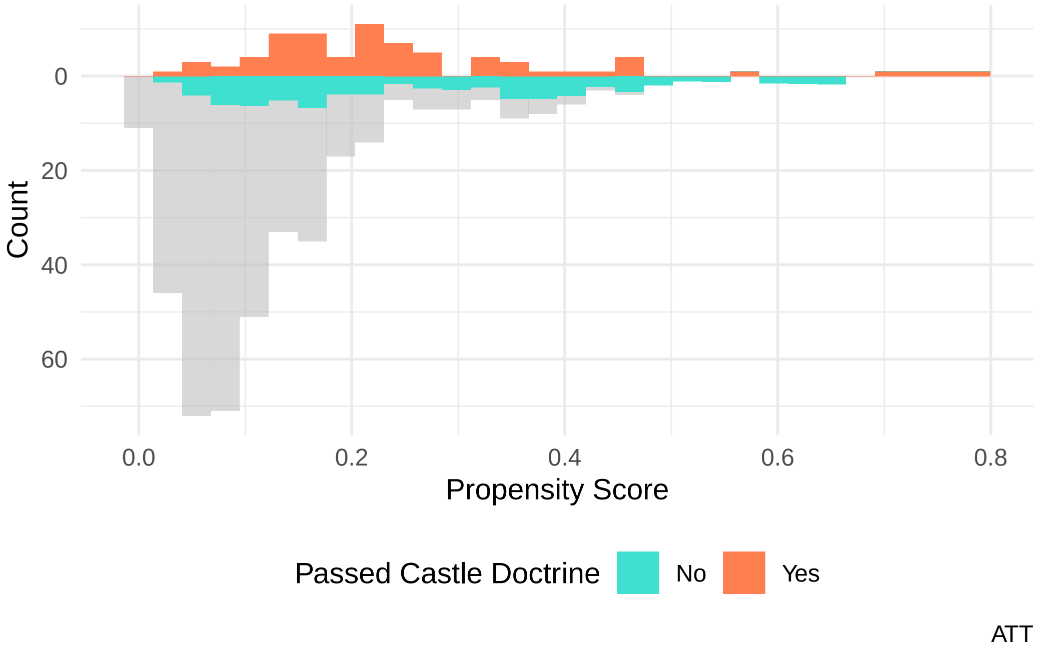
```r
p10 <- ggplot(castle_dat_test, aes(x = robbery_gun_r,
              color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Robbery with Gun", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p11 <- ggplot(castle_dat_test, aes(x = motor, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Motor", color = "Castle Implemented") +
  theme_minimal()

p12 <- ggplot(castle_dat_test, aes(x = motor, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Motor", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

(p1+p2)/(p3+p4)+ plot_layout(guides = "collect") &
      theme(legend.position = "bottom")
(p5+p6)/(p7+p8)+ plot_layout(guides = "collect") &
      theme(legend.position = "bottom")
(p9+p10)/(p11+p12)+ plot_layout(guides = "collect") &
      theme(legend.position = "bottom")


#eCDFs with splines

p1 <- ggplot(castle_dat, aes(x = burglary, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Burglary", color = "Castle Implemented") +
  theme_minimal()

p2 <- ggplot(castle_dat, aes(x = burglary, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Burglary", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p3 <- ggplot(castle_dat, aes(x = assault, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Assault", color = "Castle Implemented") +
  theme_minimal()

p4 <- ggplot(castle_dat, aes(x = assault, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Assault", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p5 <- ggplot(castle_dat, aes(x = robbery, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Robbery", color = "Castle Implemented") +
  theme_minimal()

p6 <- ggplot(castle_dat, aes(x = robbery, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Robbery", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p7 <- ggplot(castle_dat, aes(x = homicide, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Homicide", color = "Castle Implemented") +
  theme_minimal()

p8 <- ggplot(castle_dat, aes(x = homicide, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Homicide", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p9 <- ggplot(castle_dat, aes(x = robbery_gun_r, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Robbery with Gun", color = "Castle Implemented") +
  theme_minimal()

p10 <- ggplot(castle_dat, aes(x = robbery_gun_r,
            color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Robbery with Gun", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

p11 <- ggplot(castle_dat, aes(x = motor, color = factor(post))) +
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Motor", color = "Castle Implemented") +
  theme_minimal()

p12 <- ggplot(castle_dat, aes(x = motor, color = factor(post))) +
  geom_ecdf(aes(weights = w_att)) +
  theme(legend.position = "bottom") +
  labs(x = "Motor", color = "Castle Implemented") +
  ggtitle("ATT") +
  theme_minimal()

(p1+p2)/(p3+p4)+ plot_layout(guides = "collect") &
        theme(legend.position = "bottom")
(p5+p6)/(p7+p8)+ plot_layout(guides = "collect") &
        theme(legend.position = "bottom")
(p9+p10)/(p11+p12)+ plot_layout(guides = "collect") &
        theme(legend.position = "bottom")


# Alt Adjustment Histograms
```

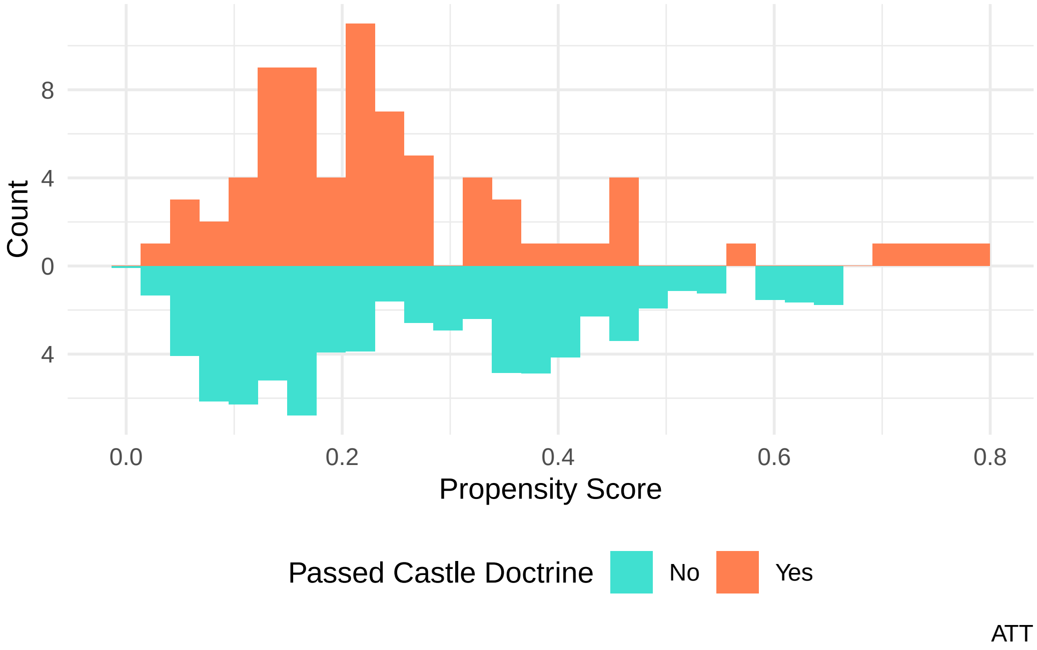
```r
p1 <- ggplot(castle_dat_alt,
            aes(x = .fitted,
            group = post,
            fill = post)) +
  geom_mirror_histogram(bins = 30, alpha = .6, fill = "grey") +
  geom_mirror_histogram(bins = 30, alpha = 1,
                    aes(fill = factor(post),
                    weight = w_att)) +
  labs(x = "Propensity Score",
      y = "Count",
      fill = "Passed Castle Doctrine",
      caption = "Mirrored Histogram of Propensity Scores") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                    values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")

p2 <- ggplot(castle_dat_alt, aes(x = .fitted,
            group = post, fill = post)) +
  geom_mirror_histogram(bins = 30, alpha = 1,
                      aes(fill = factor(post),
                      weight = w_att)) +
  labs(x = "Propensity Score",
      y = "Count",
      fill = "Passed Castle Doctrine",
      caption = "Mirrored Histogram of Propensity Scores") +
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"),
                  values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")

(p1 + p2) + plot_layout(guides = "collect") &
  theme(legend.position = "bottom")


#Alt Adjustment Love Plot

weighted_for_love_alt <- tidy_smd(
  castle_dat_alt,
  .vars = c(poverty, robbery,
            robbery_gun_r, whitem_15_24,
            blackm_15_24, whitem_25_44, blackm_25_44),
  .group = post,
  .wts = c(w_att)
)

ggplot(data = weighted_for_love_alt,
      aes(x = abs(smd), y = variable,
      group = method, color = method)) +
  geom_love() +
  scale_color_manual(values = c("coral", "turquoise"),
                labels = c("Oberserved", "ATT Weighted")) +
  labs(color = "Method",
      x = "Absolute Value of SMD",
      y = "Variable") +
  theme_minimal()
```

| Variable | Missing (n) | Percent Missing |
|---|---|---|
| robbery_gun_r | 5 | 1 |
| year | 0 | 0 |
| sid | 0 | 0 |
| jhcitizen_c | 0 | 0 |
| jhpolice_c | 0 | 0 |
| homicide | 0 | 0 |
| robbery | 0 | 0 |
| assault | 0 | 0 |
| burglary | 0 | 0 |
| larceny | 0 | 0 |
| motor | 0 | 0 |
| murder | 0 | 0 |
| unemployrt | 0 | 0 |
| blackm_15_24 | 0 | 0 |
| whitem_15_24 | 0 | 0 |
| blackm_25_44 | 0 | 0 |
| whitem_25_44 | 0 | 0 |
| poverty | 0 | 0 |
| l_homicide | 0 | 0 |
| l_larceny | 0 | 0 |
| l_motor | 0 | 0 |
| l_police | 0 | 0 |
| l_income | 0 | 0 |
| l_prisoner | 0 | 0 |
| l_lagprisoner | 0 | 0 |
| l_exp_subsidy | 0 | 0 |
| l_exp_pubwelfare | 0 | 0 |
| post | 0 | 0 |
| popwt | 0 | 0 |
| treat_year | 0 | 0 |
| murder_lag | 0 | 0 |
| state | 0 | 0 |
| pop | 0 | 0 |

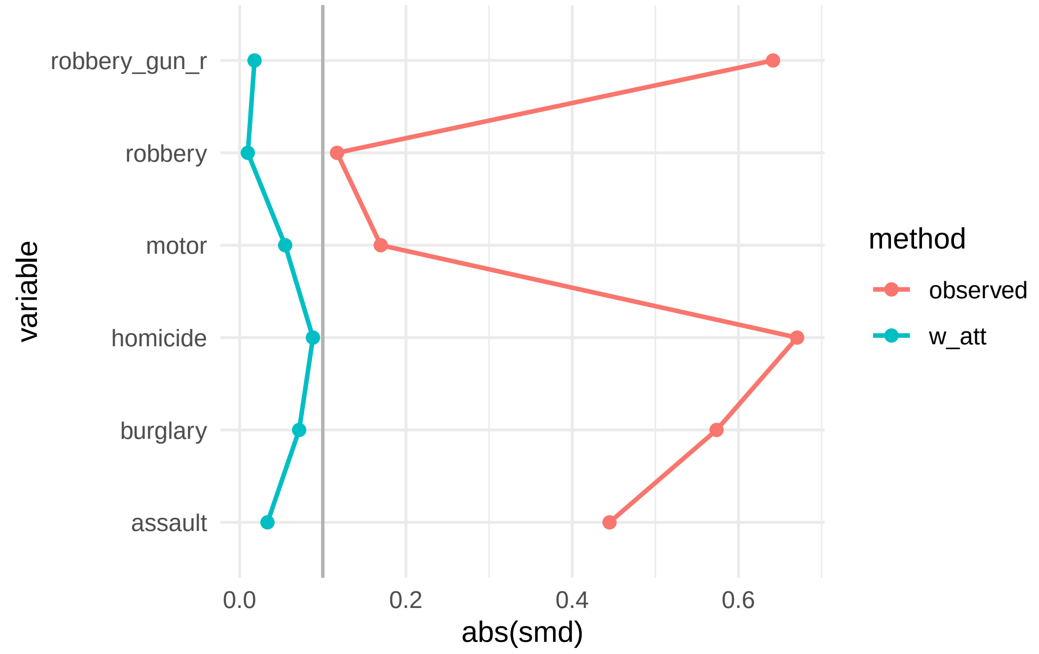
## Model before the splines:

### Mirrored Histogram:

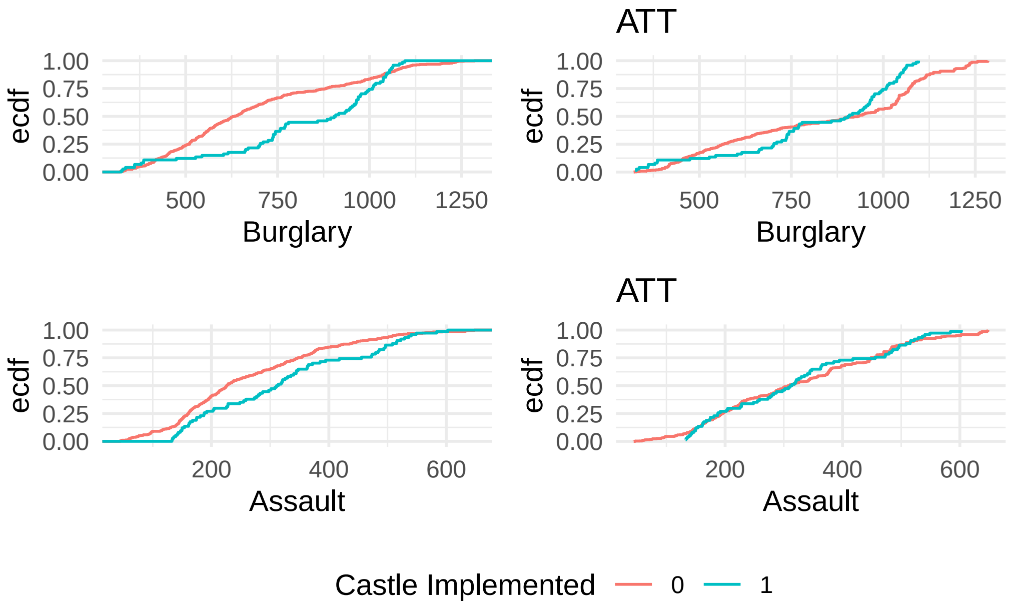


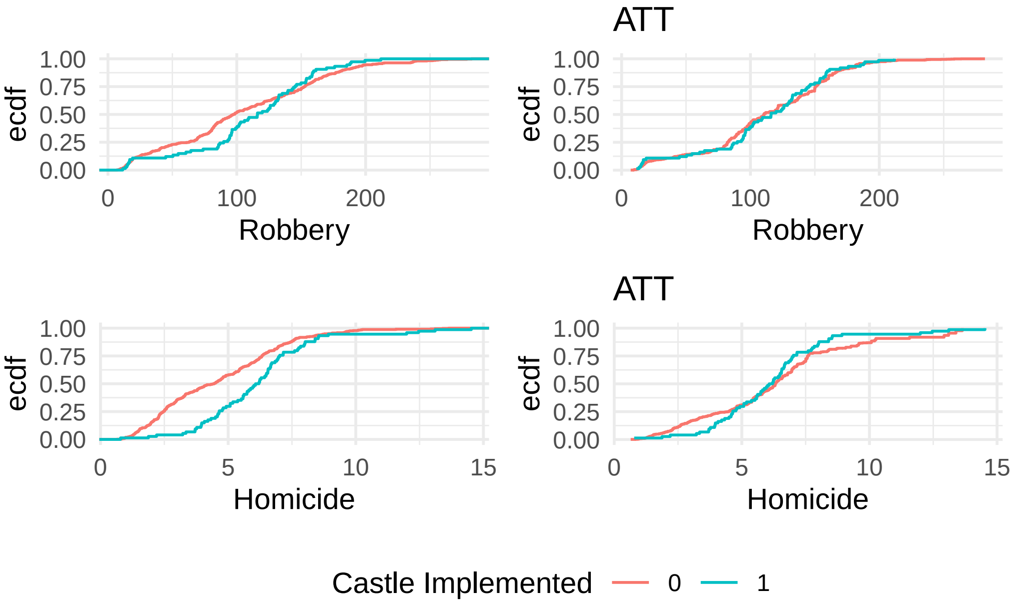


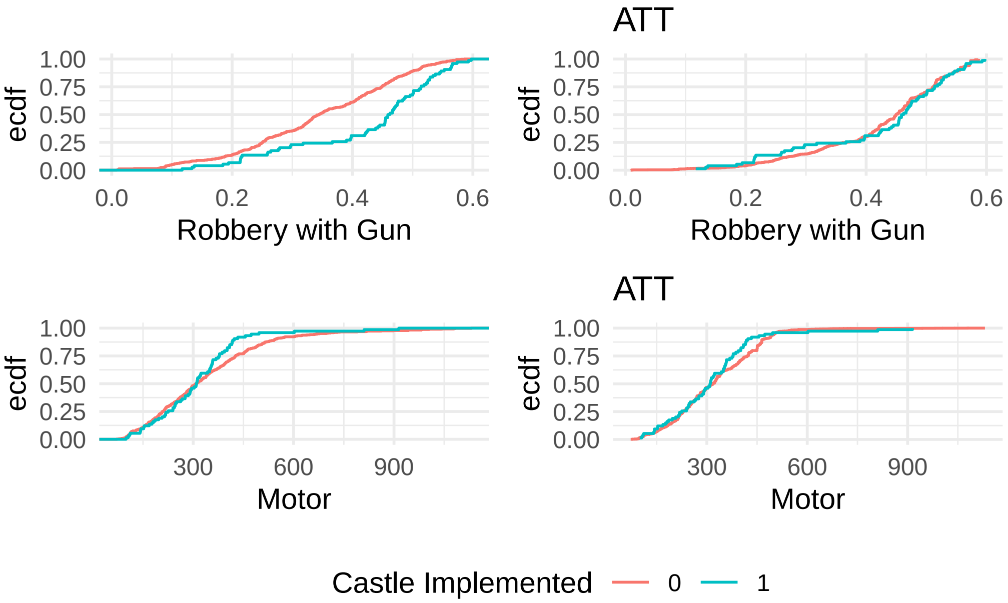
**Love Plot:**



**eCDFs:**

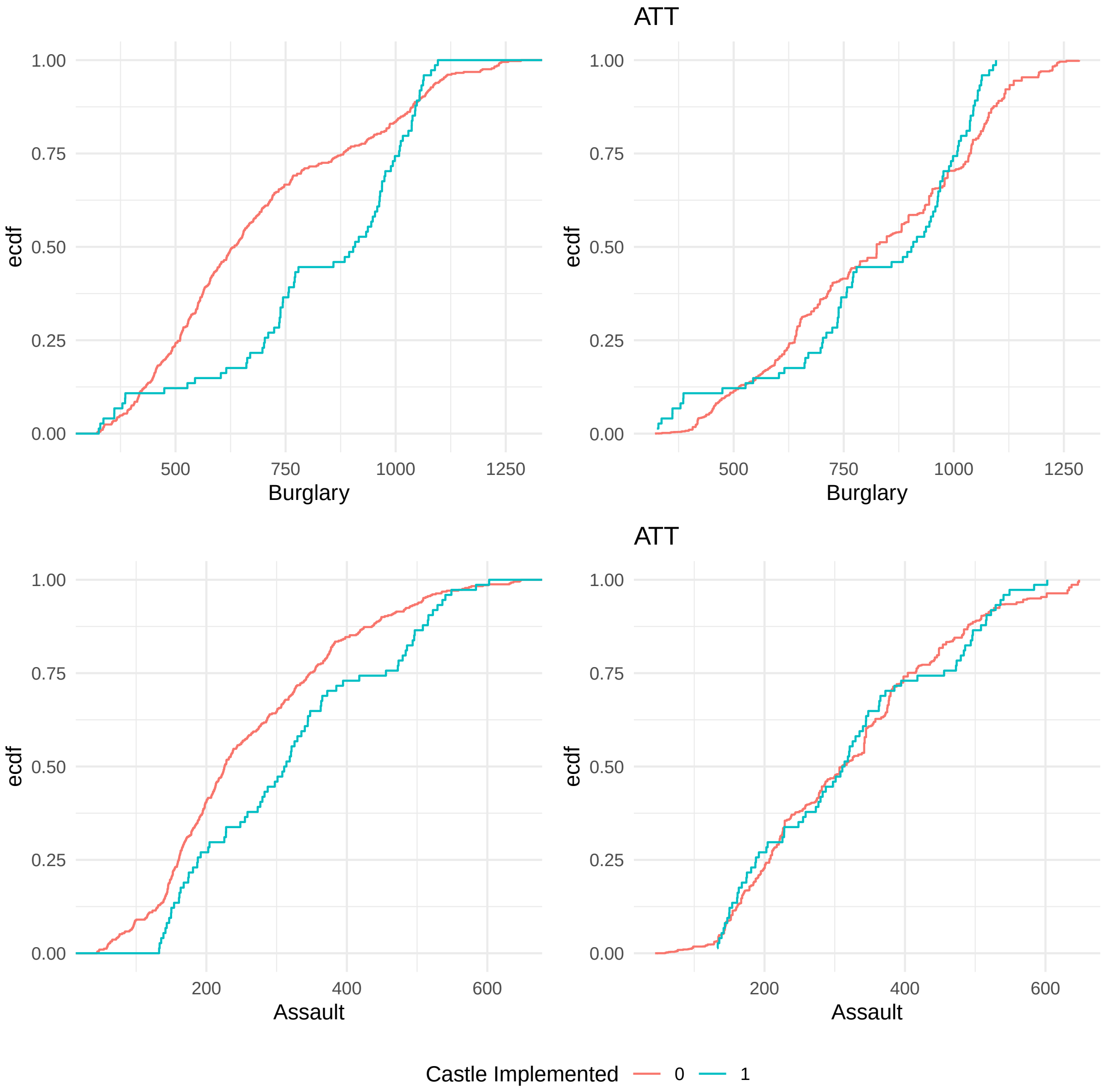


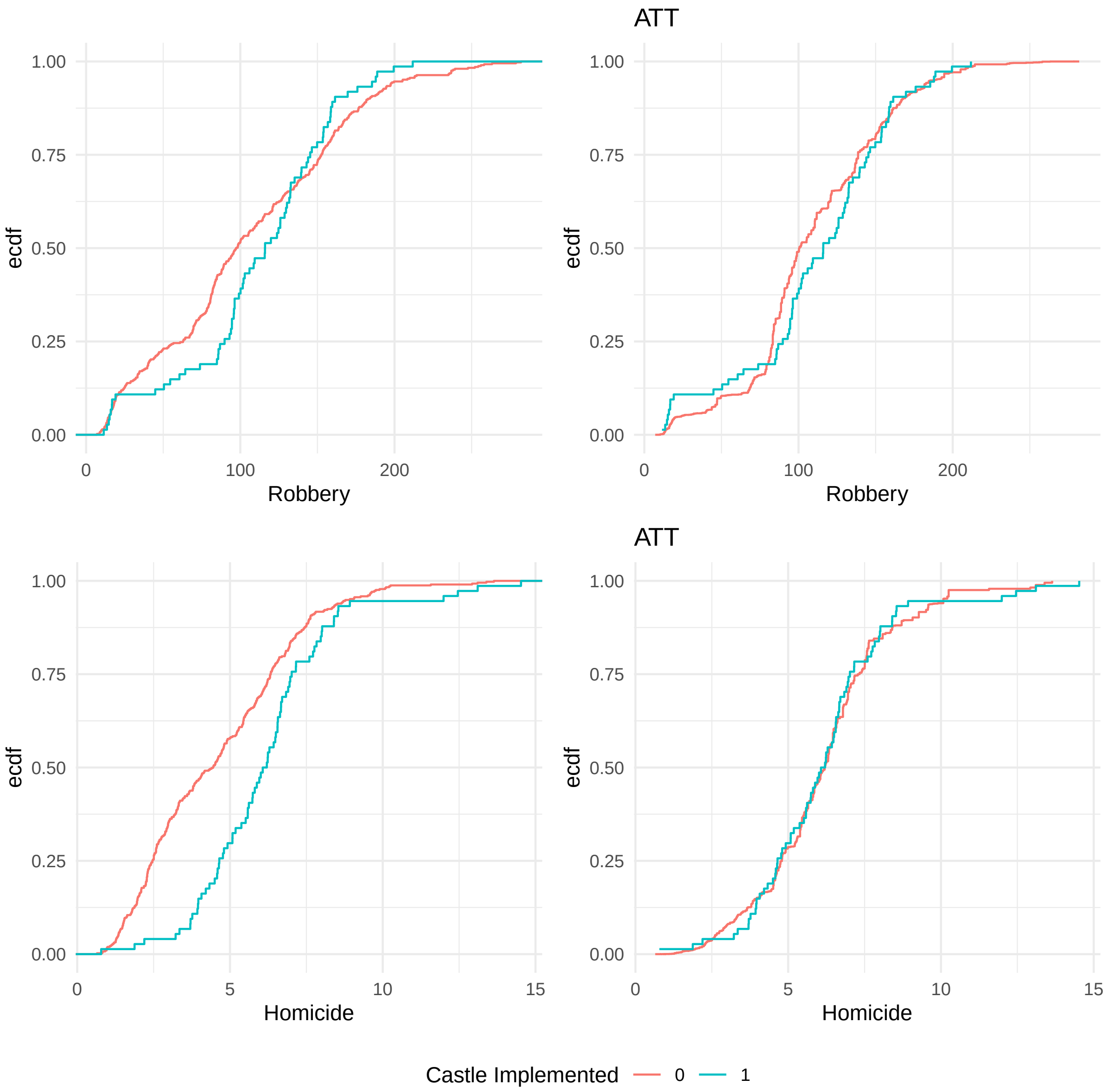

ATT
ATT
ecdf
1.00
0.75
0.50
0.25
0.00
0
100
200
Robbery
0
5
10
15
Homicide
Castle Implemented
0
1

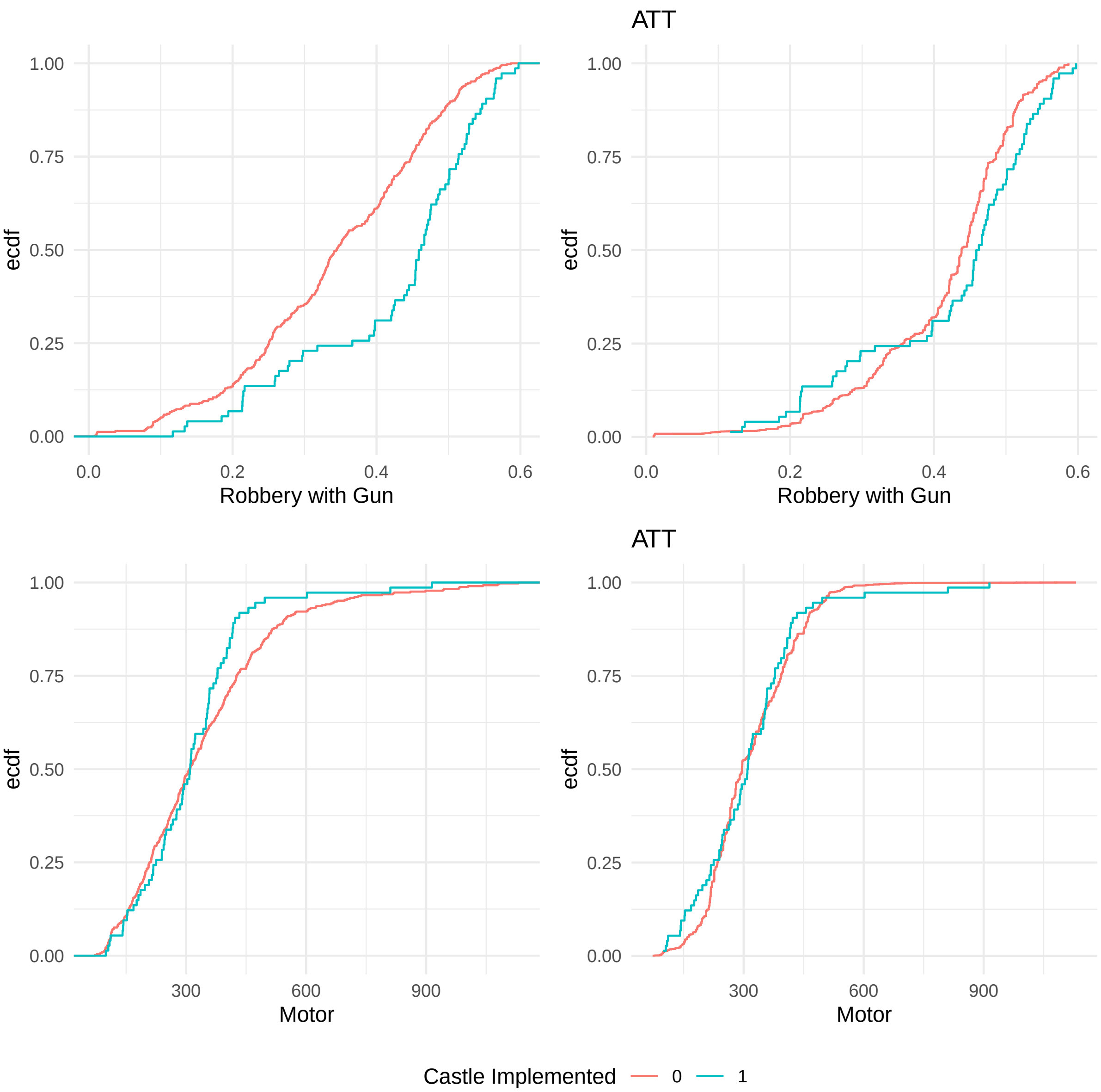

ATT
ATT
ecdf
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Robbery with Gun
300
600
900
Motor
Castle Implemented
0
1

## With splines:

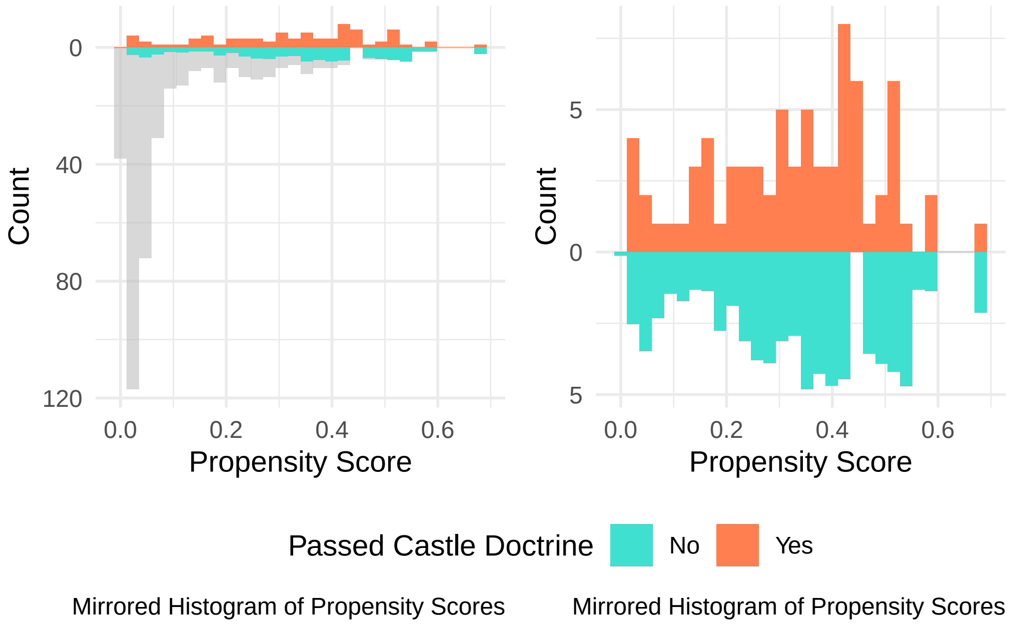
### eCDFS:

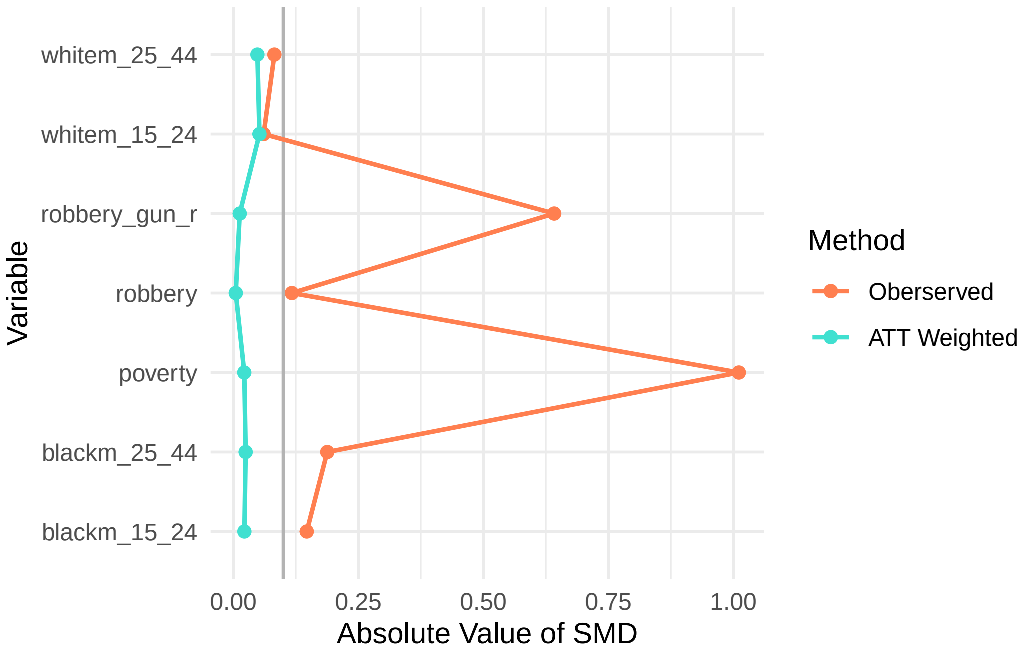



ATT
ecdf
1.00
0.75
0.50
0.25
0.00
0
100
200
Robbery
ATT
Homicide
0
5
10
15
Castle Implemented
0
1


ATT
ecdf
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Robbery with Gun
ATT
300
600
900
Motor
Castle Implemented
0
1

## Alternate Adjustment Set:


Count
Propensity Score
Count
Propensity Score
Passed Castle Doctrine
No
Yes
Mirrored Histogram of Propensity Scores
Mirrored Histogram of Propensity Scores


Variable
whitem_25_44
whitem_15_24
robbery_gun_r
robbery
poverty
blackm_25_44
blackm_15_24
Absolute Value of SMD
Method
Oberserved
ATT Weighted