Final Project

Hugh Ford and Lauren Chandler-Holtz

This is an abstract

Libraries

Missing Data Visualization

Missing Data Visualization

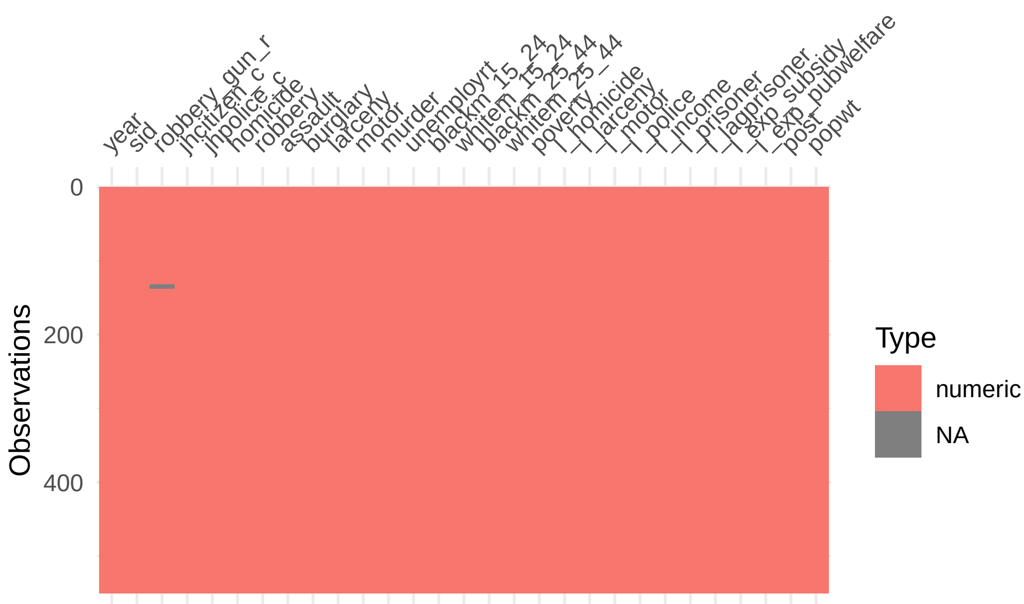


Figure 1: Missing Data Visualization

Variable	Missing (n)	Percent Missing
robbery_gun_r	6	1.09
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

Data Transformations and Cleaning

DAG

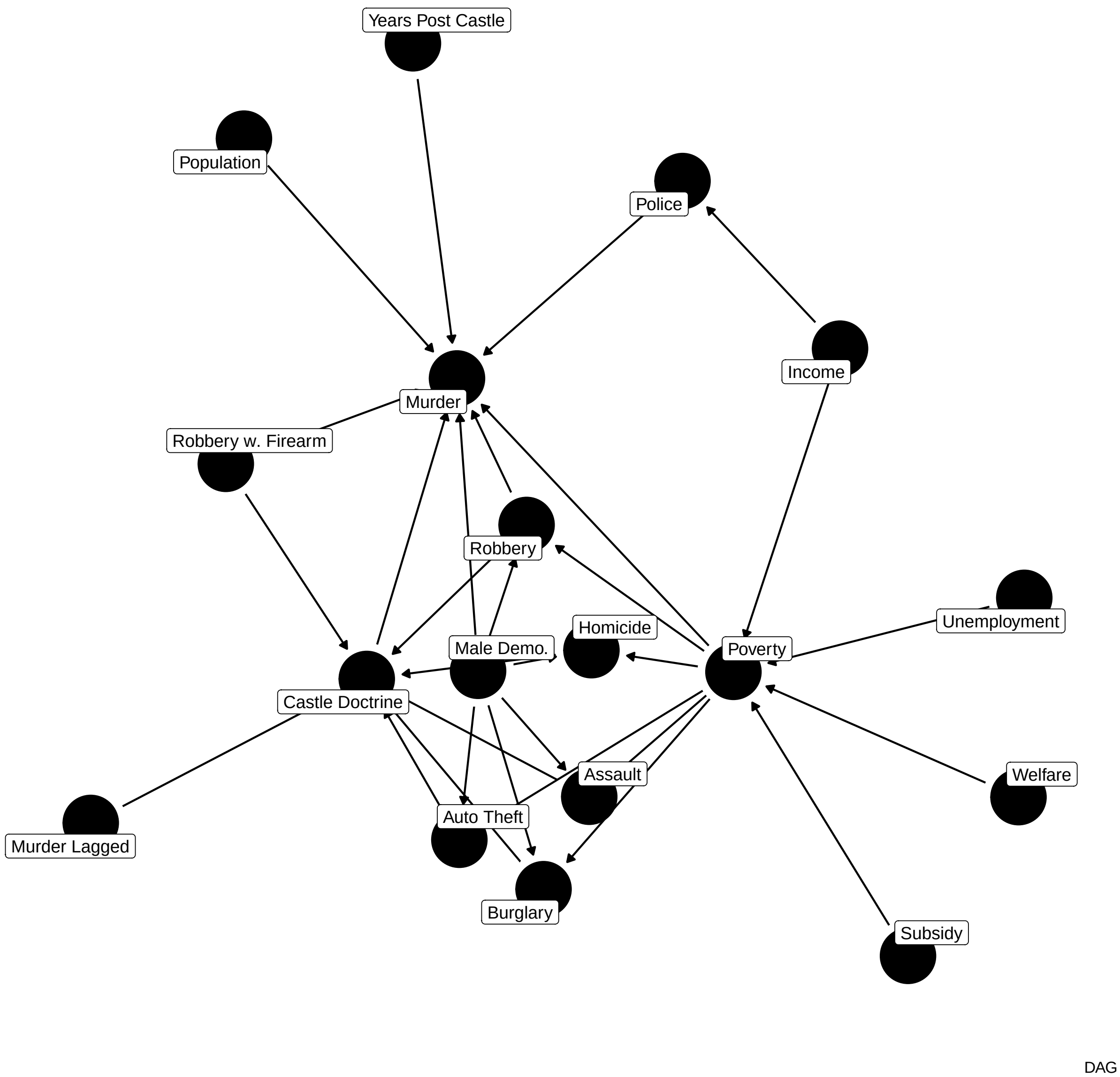


Figure 2: DAG

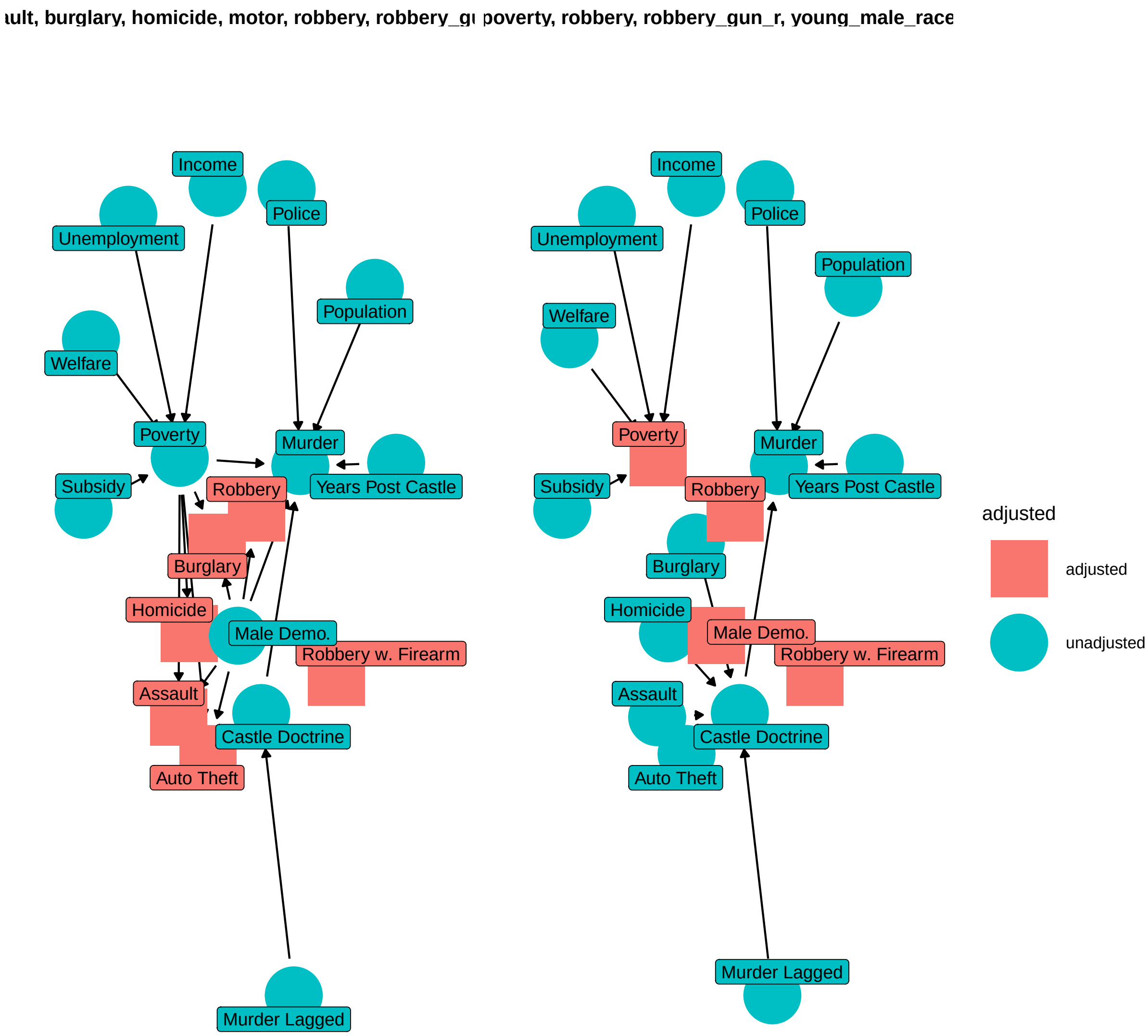
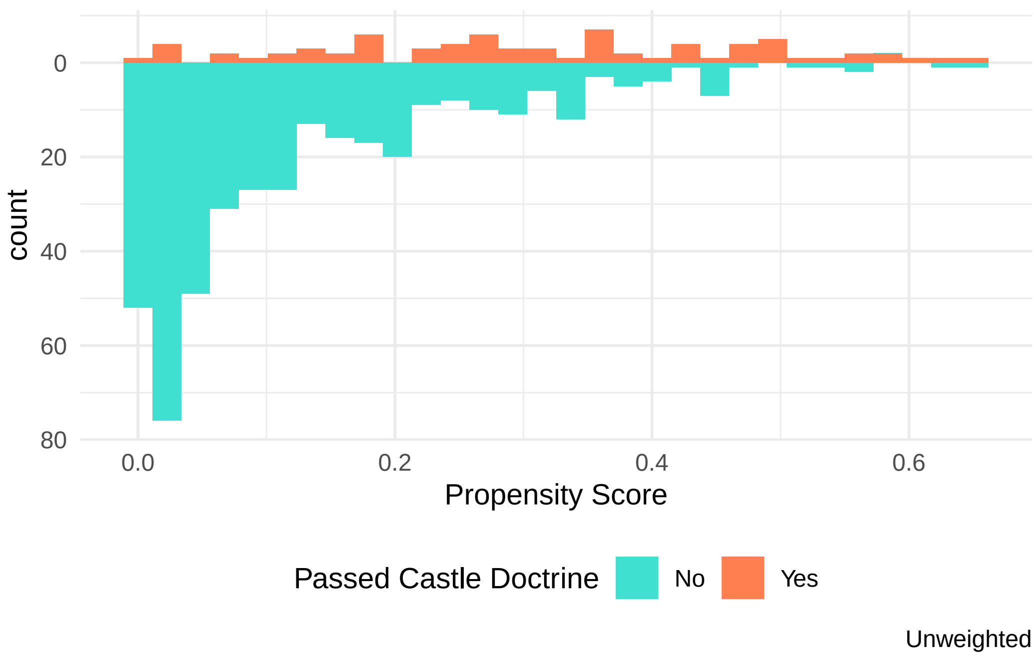


Figure 3: DAG w/ Adjustments

Characteristic[1]	**Post-Doctrine** N = 74[1]	**Pre-Doctrine** N = 411[1]	**Overall** N = 485[1]
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]Rates per 100,000 persons

Propensity Weighting



Characteristic[1]	**Post-Doctrine** N = 74; ESS = 74.0[1]	**Pre-Doctrine** N = 411; ESS = 143.7[1]
Assault	311 (188, 456)	310 (207, 404)
Burglary	903 (703, 1,008)	825 (639, 1,035)
Homicide	6.07 (4.65, 7.02)	6.19 (4.79, 7.34)
Motor Vehicle Theft	309 (225, 378)	296 (234, 394)
Robbery	116 (90, 145)	101 (83, 139)
Armed Robbery	0.46 (0.37, 0.51)	0.44 (0.36, 0.49)

[1]Rates per 100,000 persons

Abbreviation: ESS = Effective Sample Size

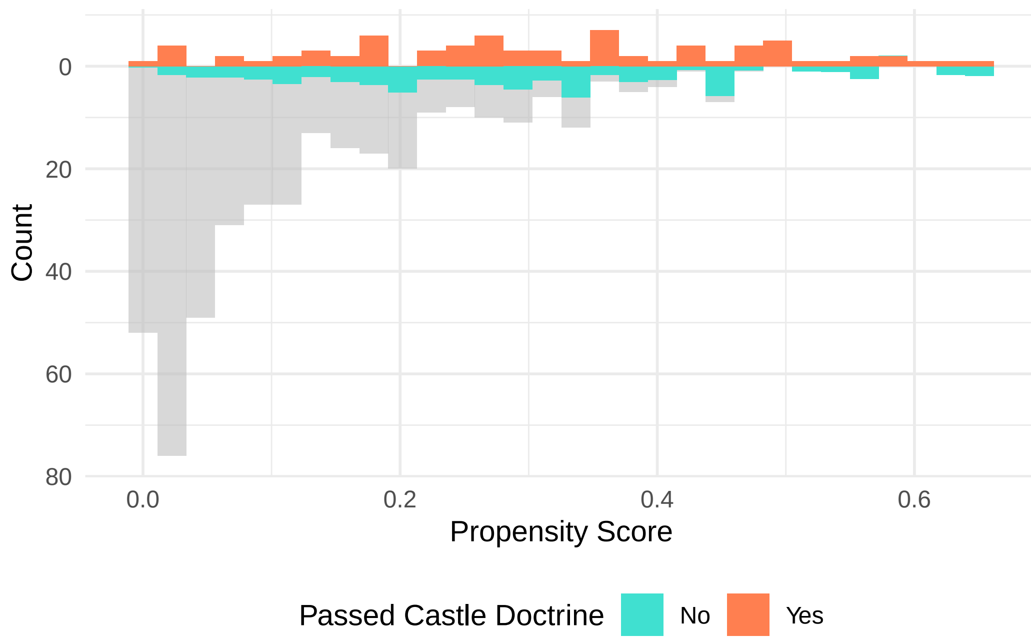


Figure 4: Mirrored Histogram, Unweighted and Weighted

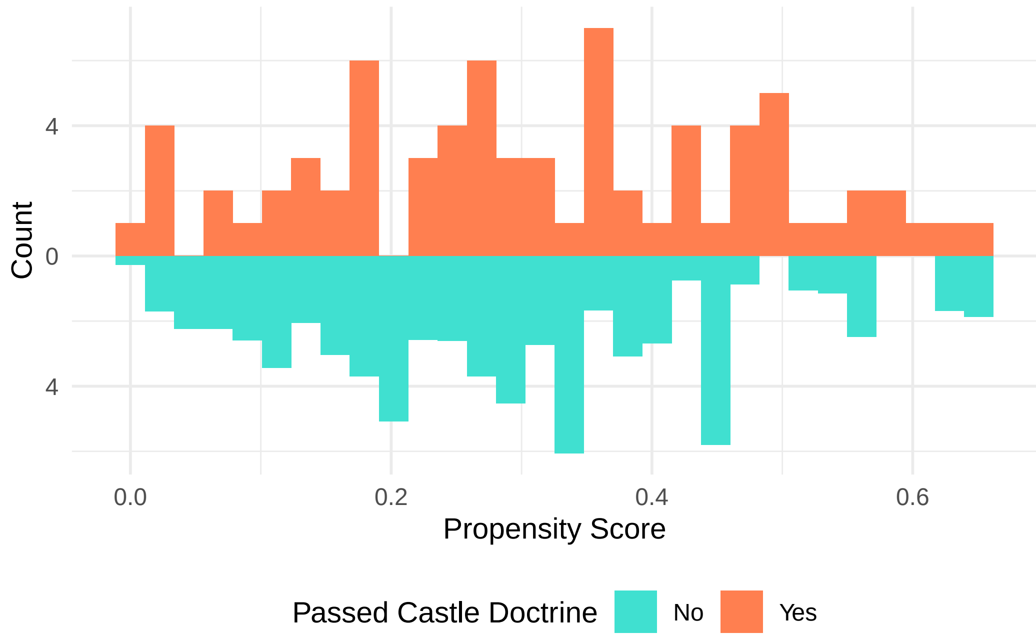


Figure 5: Mirrored Histogram, ATT Only

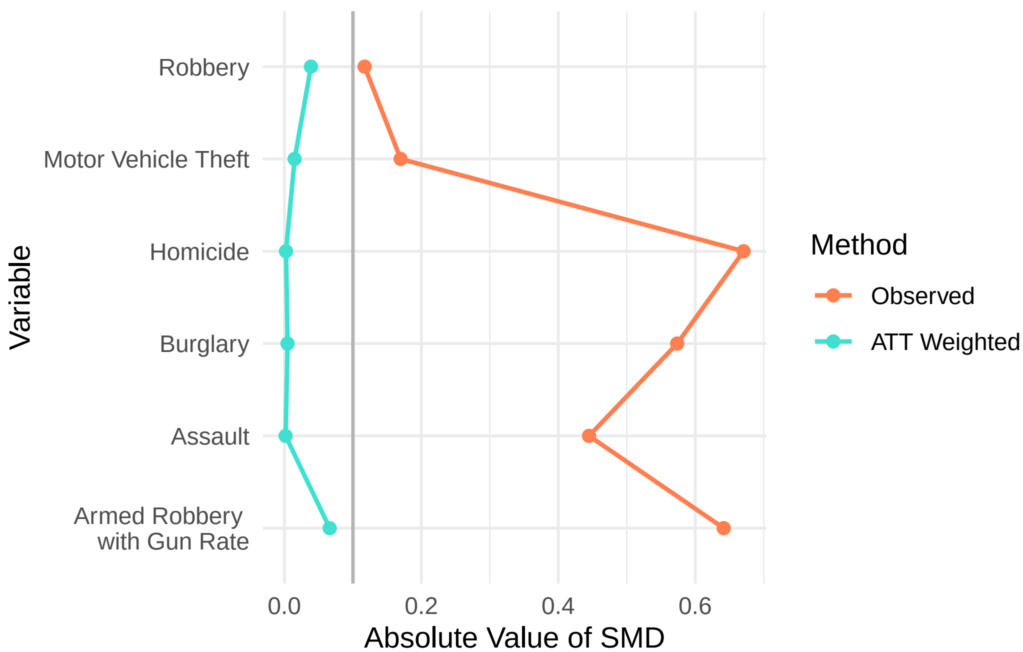


Figure 6: Love Plot

Years Before After Castle Doctrine	ATT Point Est.
1	-0.797
2	-1.593
3	-2.390
4	-3.186

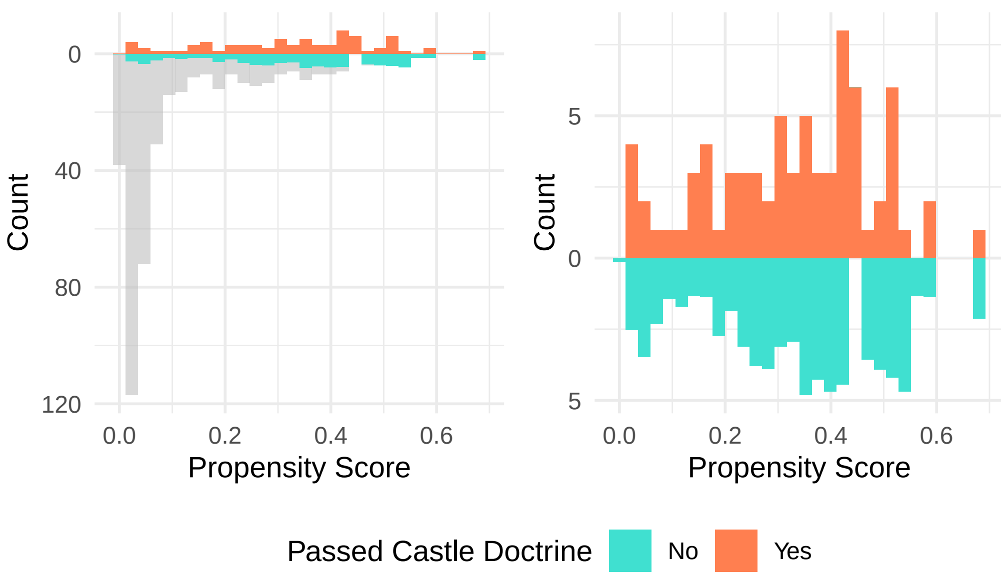
Years Before After Castle Doctrine	ATT Estimate	ATT Standard Dev.	95% CI Lower Bound	Up
1	-0.781	0.584	-1.881	
2	-1.546	1.137	-3.847	
3	-2.323	1.737	-5.580	
4	-3.110	2.370	-7.829	

G-Computation

Uncertainty:

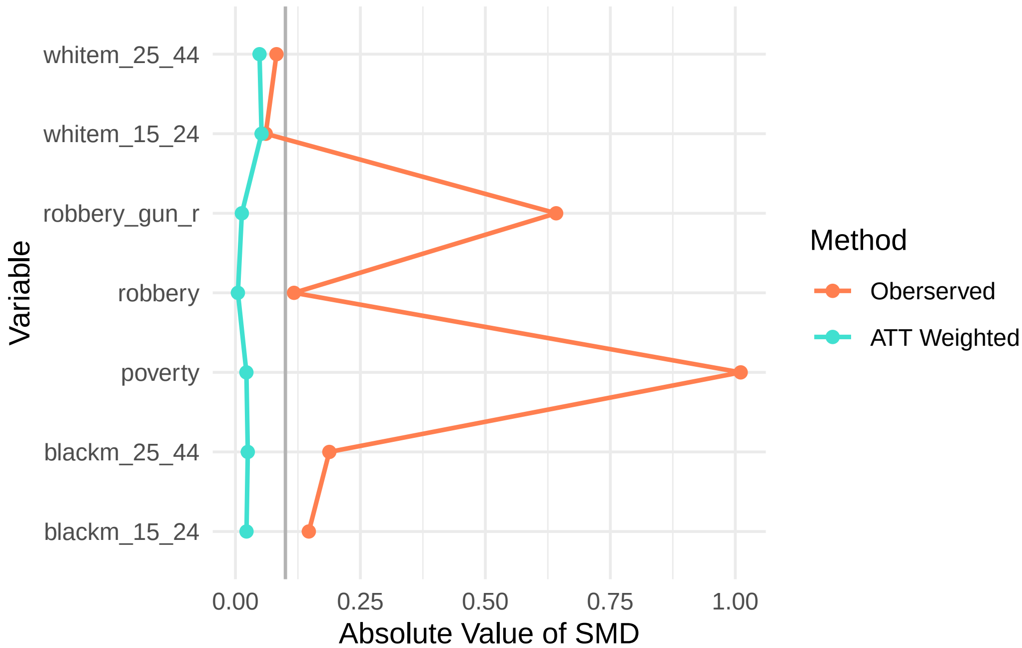
Sensitivity Analysis:

Alternate Adjustment Set:



Mirrored Histogram of Propensity Scores Mirrored Histogram of Propensity Scores

Years Before After Castle Doctrine	ATT Estimate	ATT Standard Dev.	95% CI Lower Bound	Up
1	-0.781	0.584	-1.881	
2	-1.546	1.137	-3.847	
3	-2.323	1.737	-5.580	
4	-3.110	2.370	-7.829	




```
  years_pre_post    ATT_est
1              1 -0.3937709
2              2 -0.7875419
3              3 -1.1813128
4              4 -1.5750838
```

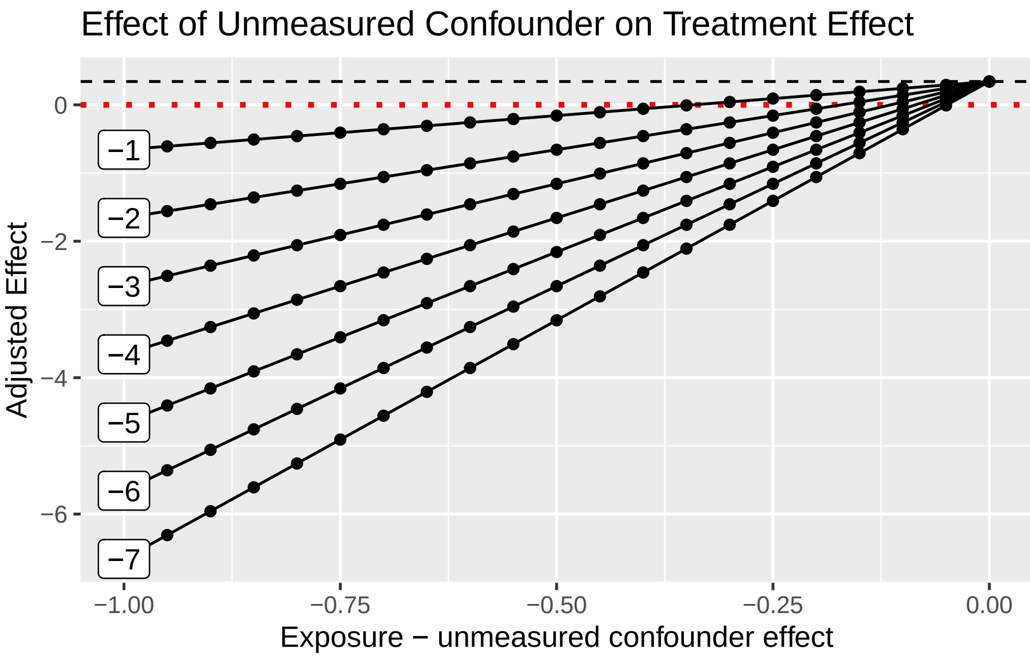
Effect direction is the same, but, especially the further out from implementation we go, the smaller the effect compared to the original DAG.

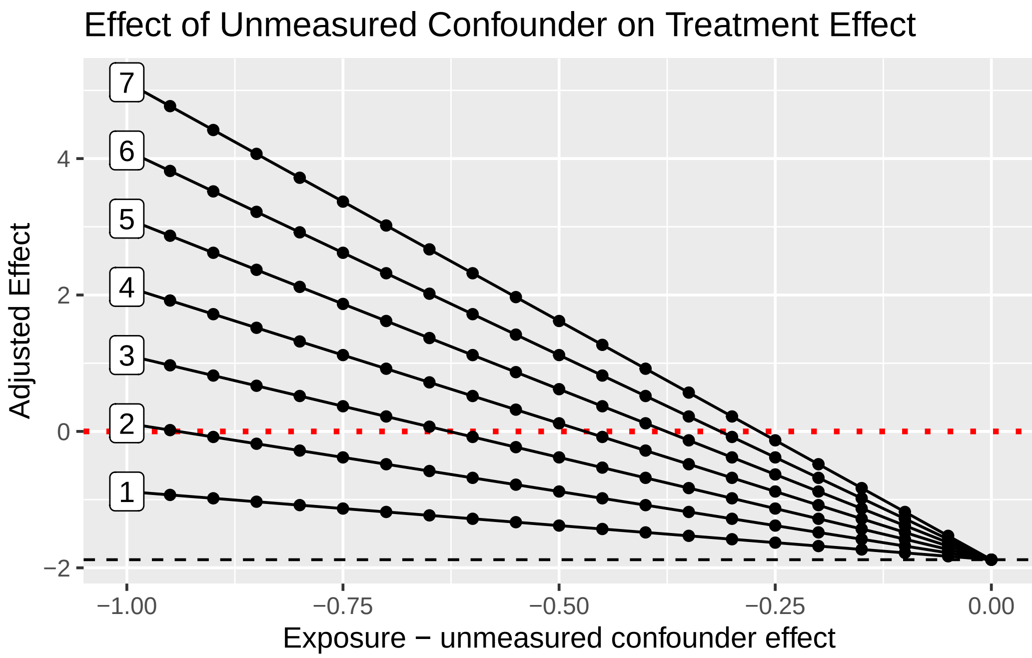
Tipping Point Analysis

Finally, we assessed the potential impact of an unmeasured confounder on the relationship between Castle Doctrine and Murder Rate. The graph below illustrates how varying strength of an unmeasured confounder's associations with the exposure (Castle

Doctrine) and the outcome (Murder Rate) would bias the estimated effect. Based on the graphs, we can identify tipping points, where the adjusted effect crosses the null.

For example, if a one standard deviation change in the unmeasured confounder reduced murder rate by 1 percent and was associated with a probability of 0.35 of not passing a Castle Doctrine law (i.e. -0.35 on the x-axis), then the upper bound of adjusted effect would cross zero, such that the 95% CI for the ATT is entirely below zero, representing a significant negative effect. Since our dataset is limited in its scope of variables, it is reasonable to believe that such a confounder might exist.






```r
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)

data("castle")
castle_for_tab <- castle |>
  select(!(starts_with("r20") |
             starts_with("trend") |
             starts_with("lead") |
```

```
            starts_with("lag")))

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

vis_dat(castle_for_tab)
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] !=
  castle$treat_year[lower:upper] <- rep(treat_year_1, 11)
  lower <- upper + 1
  upper <- lower + 10
  i <- i + 1
}

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

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


castle <- castle |>
  group_by(sid) |>
  mutate_at(c("assault", "burglary", "homicide", "larceny", "motor", "l_larceny", "l_m
              "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 |>
```

```r
  select(state, sid)
castle_dat <- full_join(castle, state_id_list, by = "sid")

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

#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)
# young_male_race is taking place of blackm_15_24 + whitem_15_24 + blackm_25_44 + whit

castle_dag <- dagify(
  murder ~ young_male_race + poverty + popwt + robbery_gun_r + l_police + post + years

  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",
```

```r
    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")
)
ggdag(castle_dag, layout = "nicely", use_labels = "label", text = FALSE) +
  labs(caption = "DAG") +
  theme_dag()
ggdag_adjustment_set(castle_dag, text_col = "black",
                     use_labels = "label",
                     text = FALSE) +
  theme_dag()
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("**Table 1: Sample Characteristics by Castle Doctrine**") |>
  modify_footnote(everything() ~ "Rates per 100,000 persons")
```

```r
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"))
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")

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("**Table 2: Sample Characteristics by Re-Weighted Castle Doctrine**")
  modify_footnote(everything() ~ "Rates per 100,000 persons")

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")

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")
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",

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

```r
ggplot(data = weighted_for_love, aes(x = abs(smd), y = variable, group = method, color
  geom_love() +
  scale_color_manual(values = c("coral", "turquoise"), labels = c("Observed", "ATT Wei
  labs(color = "Method", x = "Absolute Value of SMD", y = "Variable") +
  theme_minimal()
standardized_model <- lm(murder ~ post*years_after_treat, data = castle_dat, weights =

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 \n Castle Doctrine", "ATT Point Est.")


estimate_df |>
  gt()
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)
```

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

colnames(bootstrap_df) <- c("Years Before After \n Castle Doctrine", "ATT Estimate", "

bootstrap_df |>
  gt()


propensity_model_alt <- glm(post ~ splines::ns(poverty, 5) + robbery + splines::ns(rob

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

ggplot(data = weighted_for_love_alt, aes(x = abs(smd), y = variable, group = method, c
```

```
  geom_love() +
  scale_color_manual(values = c("coral", "turquoise"), labels = c("Oberserved", "ATT W
  labs(color = "Method", x = "Absolute Value of SMD", y = "Variable") +
  theme_minimal()
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
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 |>
      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 |>
```

```r
    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_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] <- 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_names_clean) <- c("Years Before After \n Castle Doctrine", "ATT

bootstrap_df_names_clean |>
  gt()

#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"
  )
propensity_model_test <- glm(post ~ homicide + burglary + assault + motor + robbery +

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 = "
  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 = "
  scale_y_continuous(labels = abs) +
  scale_fill_manual(labels = c("No", "Yes"), values = c("turquoise", "coral")) +
  theme_minimal() +
  theme(legend.position = "bottom")
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",
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
  geom_love() +
  theme_minimal()
p1 <- ggplot(castle_dat_test, aes(x = burglary, color = factor(post))) +
  geom_ecdf() +
```

```r
  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()

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()

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"
p1 <- ggplot(castle_dat, aes(x = burglary, color = factor(post))) +
```

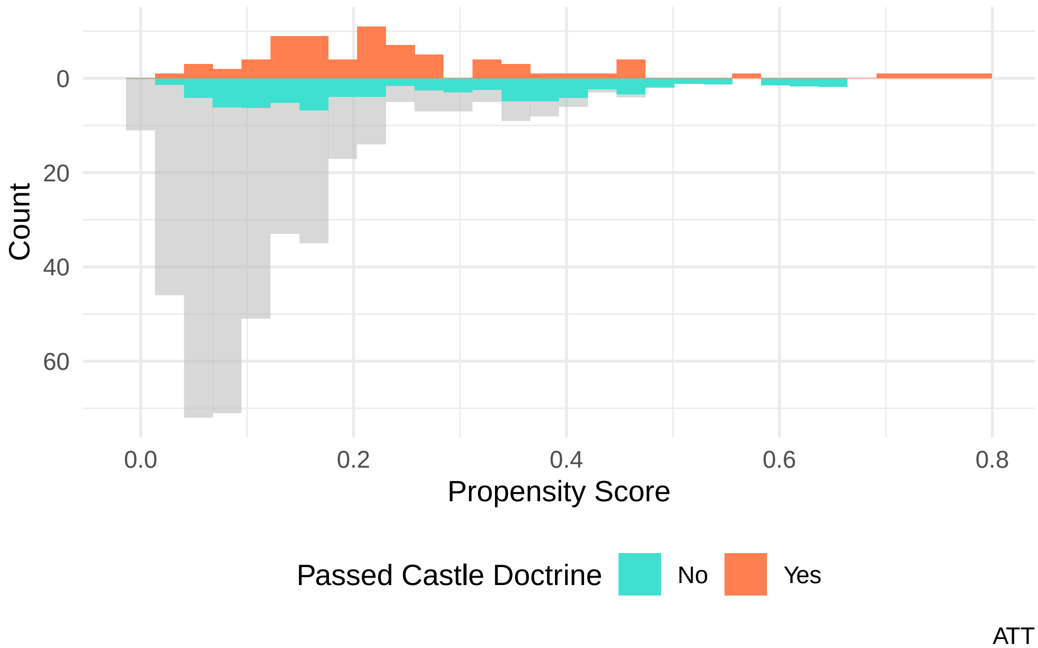
```
  geom_ecdf() +
  theme(legend.position = "bottom") +
  labs(x = "Burglary", color = "Castle Implemented") +
  theme_minimal()

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() +
```

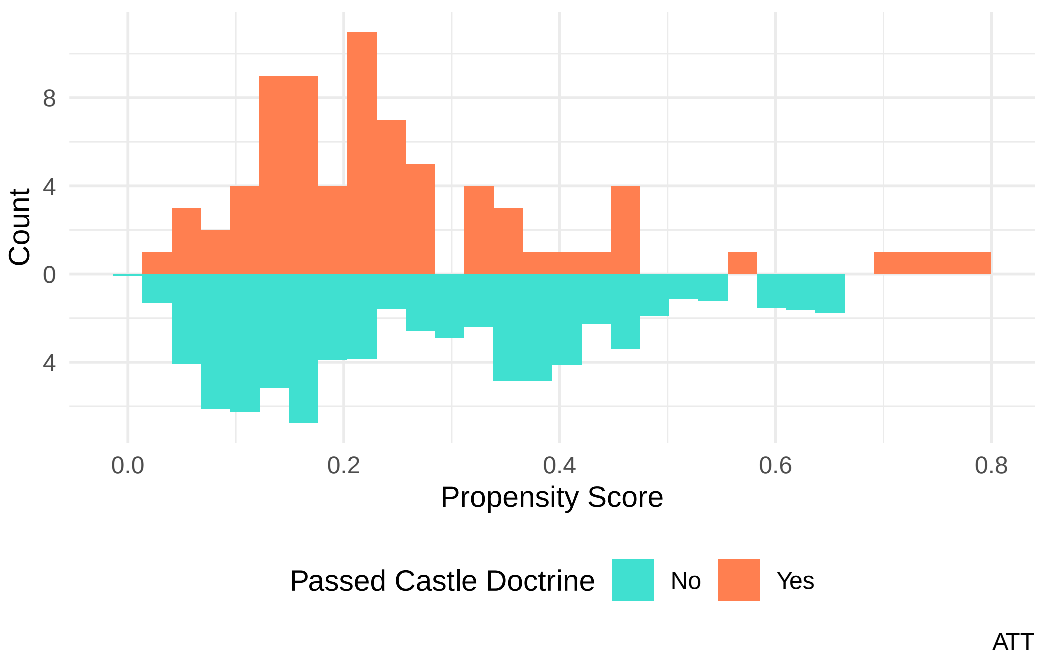
```r
  theme(legend.position = "bottom") +
  labs(x = "Homicide", color = "Castle Implemented") +
  theme_minimal()

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"
```

Appendix:

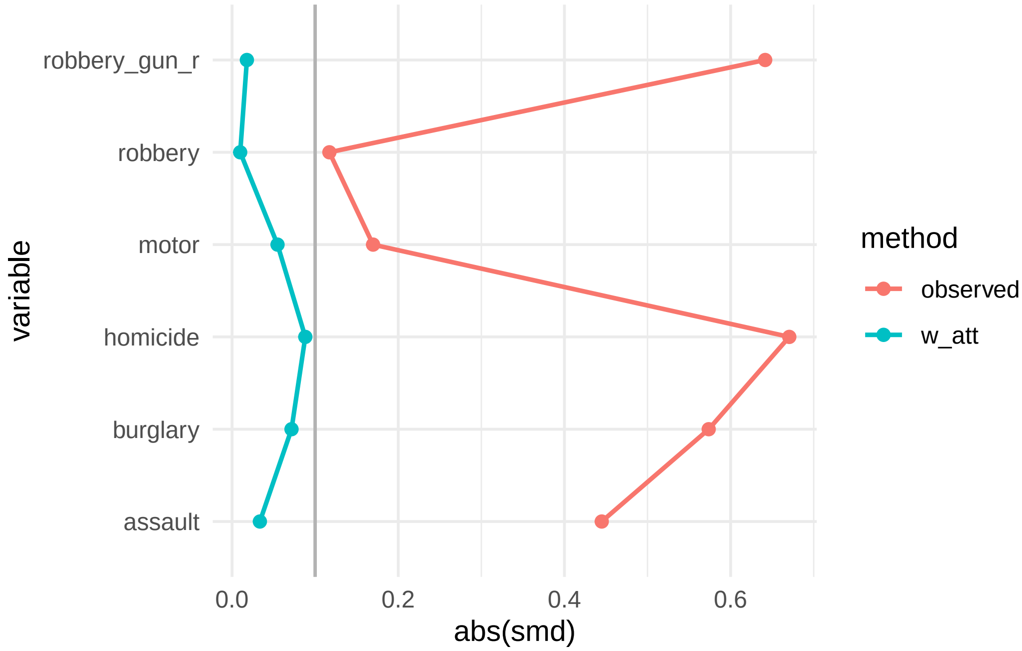
Model before the splines:

Mirrored Histogram:

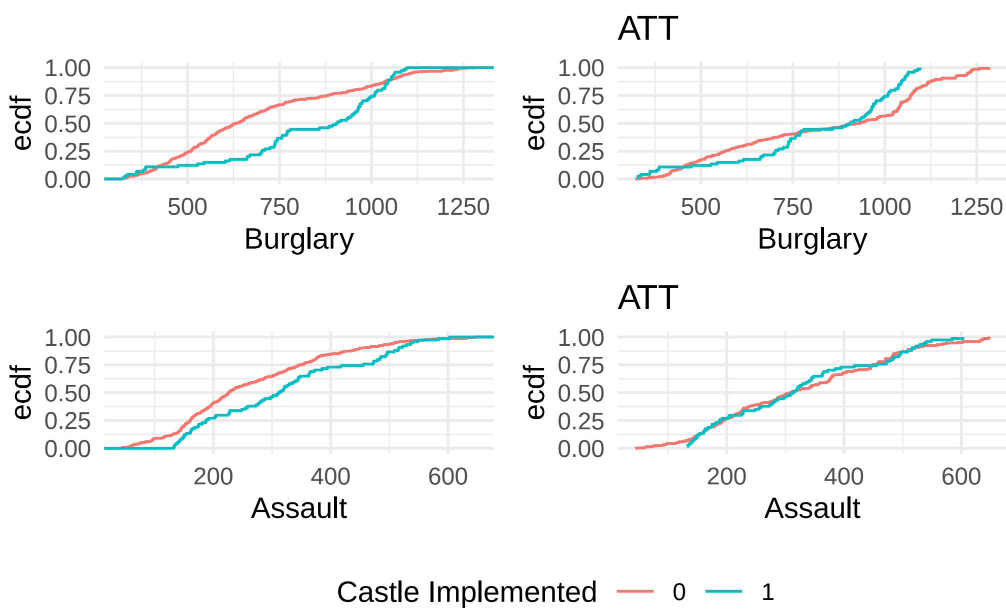


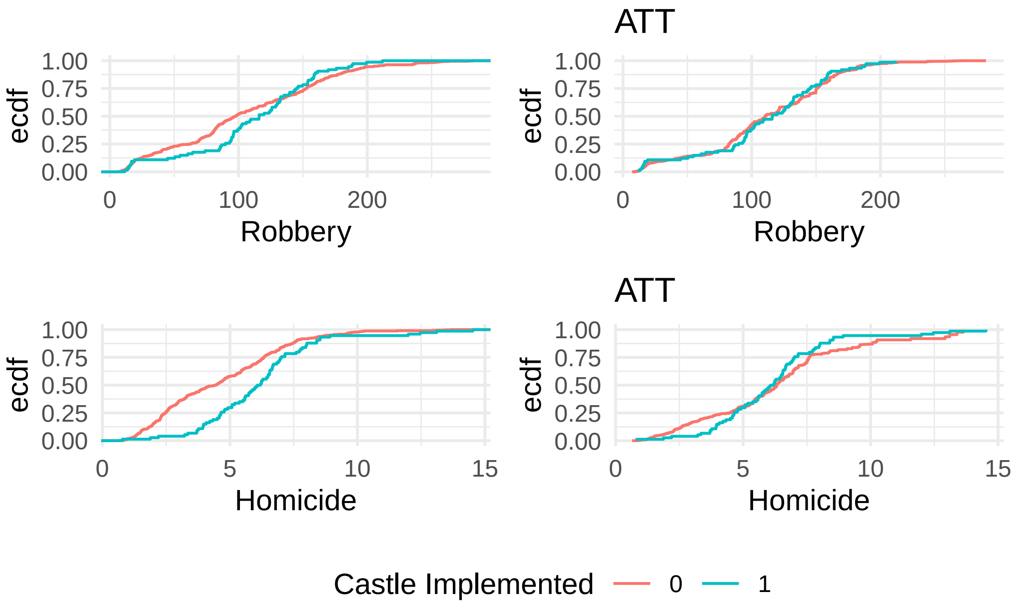


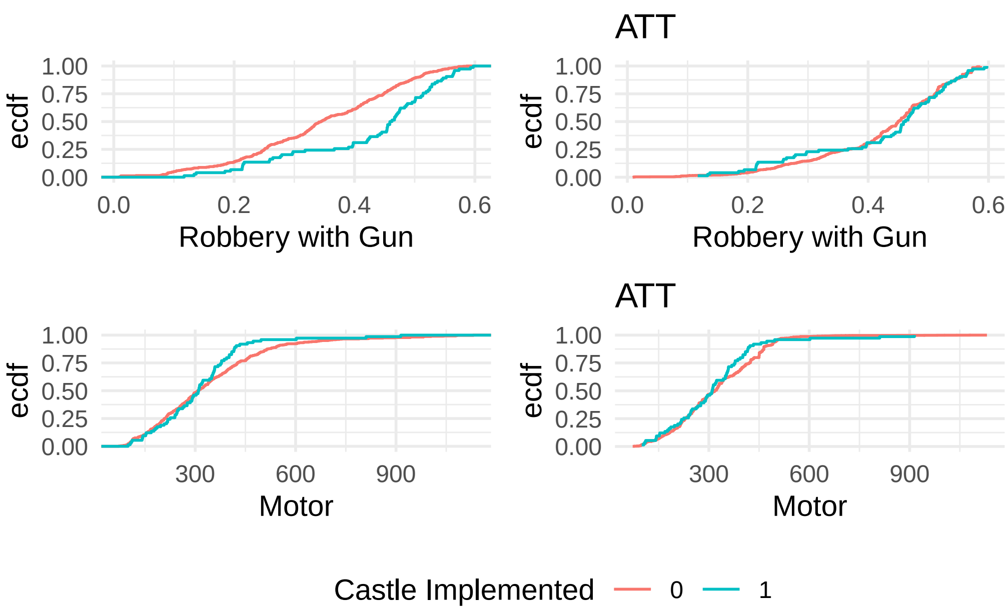
Love Plot:



eCDFs:

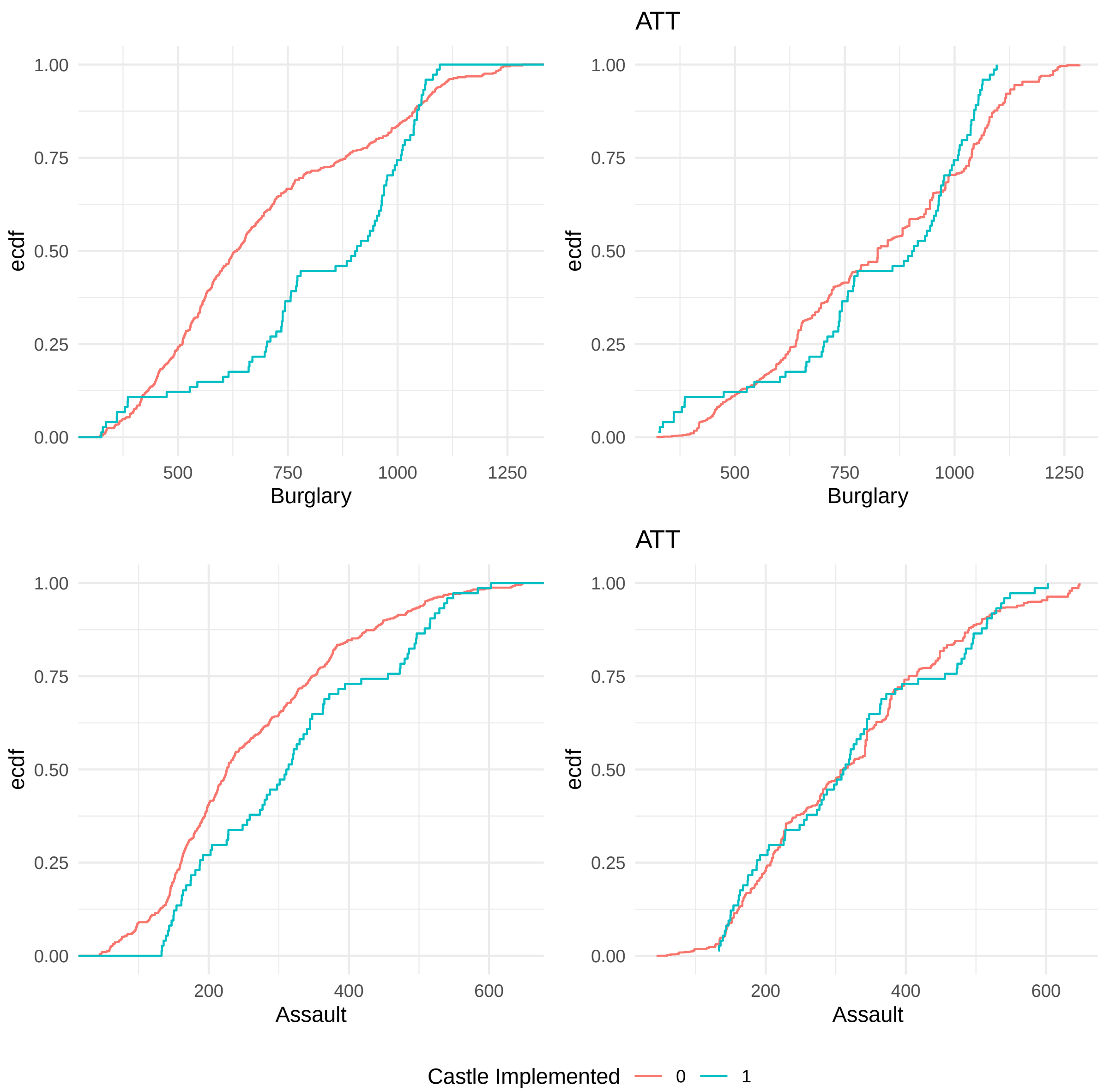


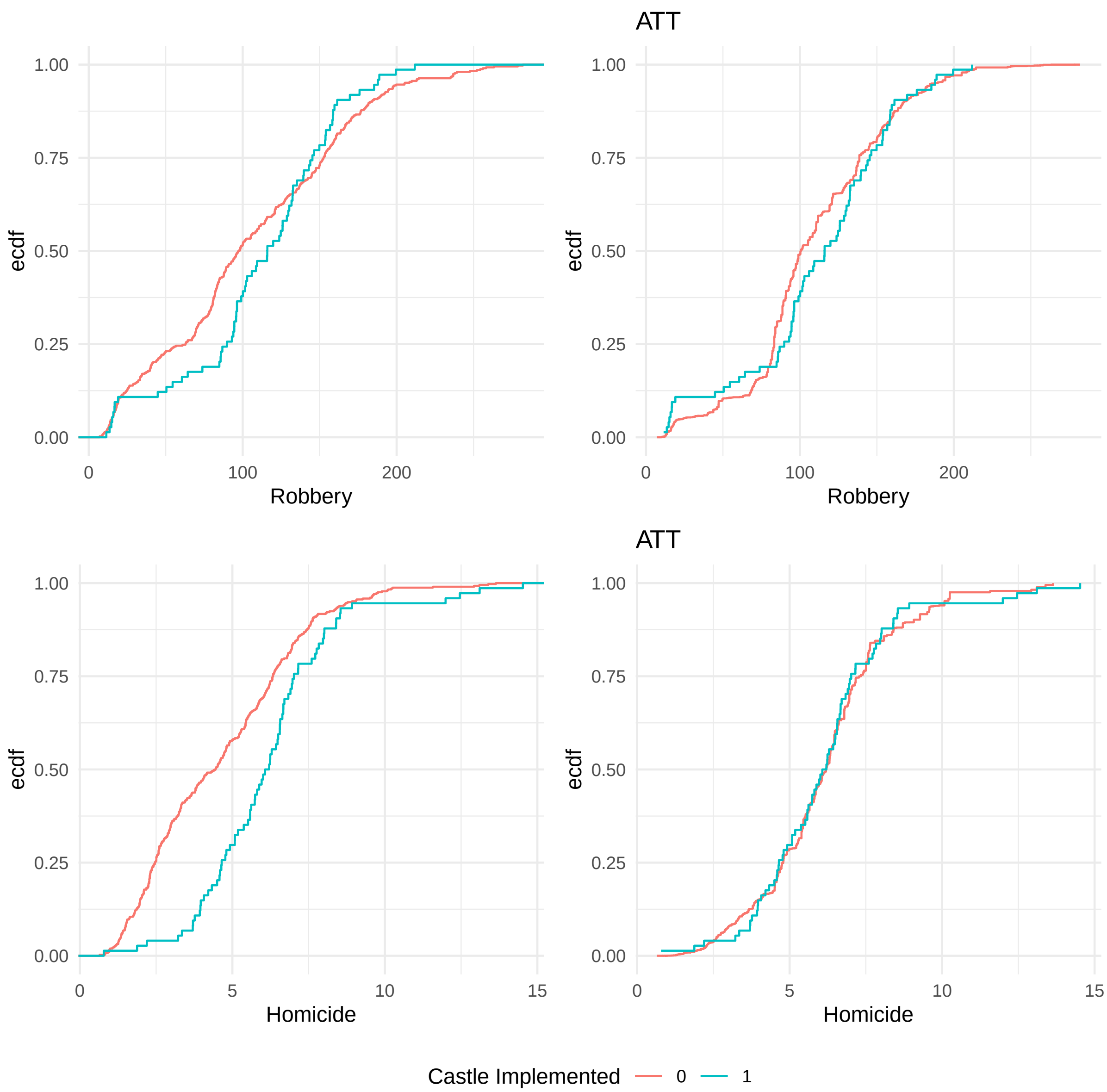

ATT
ecdf
1.00
0.75
0.50
0.25
0.00
0
100
200
Robbery
ATT
0
5
10
15
Homicide
Castle Implemented
0
1

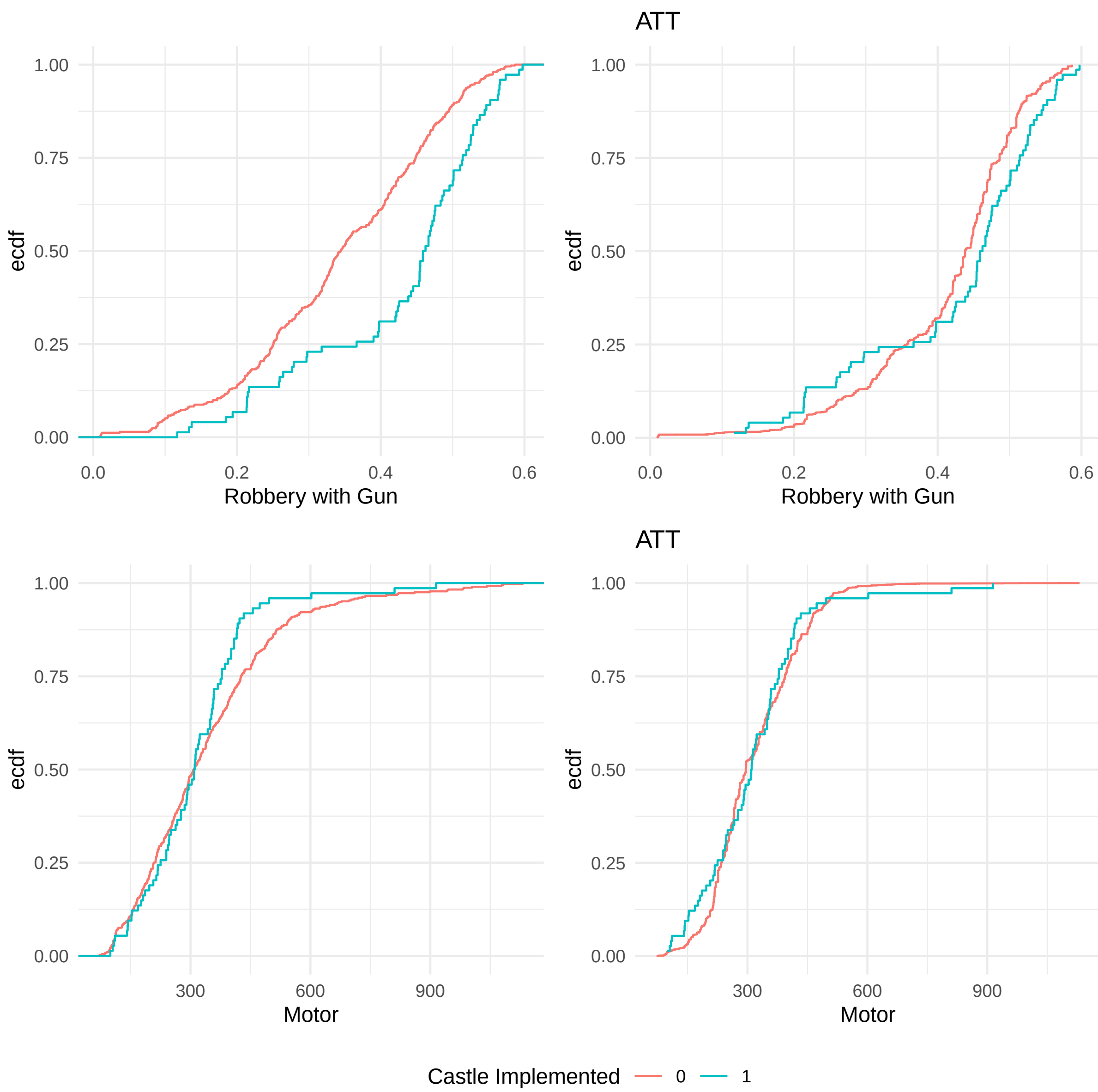

ATT
ecdf
1.00
0.75
0.50
0.25
0.00
0.0
0.2
0.4
0.6
Robbery with Gun
ATT
300
600
900
Motor
Castle Implemented
0
1

With splines:

eCDFS:




ATT
ATT
ecdf
Robbery
Homicide
Castle Implemented
0
1


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