Code 2

# Model 1C- Violent crimes with balanced auxilliary covariates- Female veterans

# Set working directory

setwd("C:/Users/victo/OneDrive/Documents/Dissertation/Psy/R3")

# Load Required Libraries

library(ggrepel)

library(magrittr)

library(dplyr)

library(augsynth)

library(tidyr)

library(ggplot2)

# Load your data

mydata <- read.csv("chapter1log.csv")

# Step 1: Extract Relevant Columns

analysis\_df <- mydata %>%

select(Year, Month, City, Violent\_per\_100k, Property\_per\_100k, Female.Veterans, Population.Density, Gini.Index..Income.Inequality.)

# Step 2: Create `Date` Variable (YYYY-MM Format)

analysis\_df <- analysis\_df %>%

mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))

# Step 3: Define Treatment Dates for Treated Cities

treatment\_dates <- data.frame(

City = c("Oakland", "Santa Cruz", "Arcata", "San Francisco", "Berkeley"),

Tdate = as.Date(c("2019-06-01", "2020-01-01", "2021-10-01", "2022-09-01", "2023-07-01"))

)

# Step 4: Merge `Tdate` into `analysis\_df`

analysis\_df <- analysis\_df %>%

left\_join(treatment\_dates, by = "City")

# Step 5: Create `Treat` Variable (Treatment Status)

analysis\_df <- analysis\_df %>%

mutate(

Tdate = ifelse(is.na(Tdate), Inf, Tdate),

Treat = ifelse(Date >= Tdate, 1, 0)

)

# Ensure correct formatting

analysis\_df <- analysis\_df %>%

mutate(Tdate = as.Date(Tdate, origin = "1970-01-01"))

# Step 6: Run the multisynth model

synth\_model <- multisynth(

Violent\_per\_100k ~ Treat | Female.Veterans + Population.Density + Gini.Index..Income.Inequality.,

unit = City,

time = Date,

data = analysis\_df

)

# ✅ \*\*Save Result 1: Nu Values\*\*

write.csv(synth\_model$nu, "1cResults\_Nu.csv", row.names = TRUE)

# ✅ \*\*Save Result 2: Model Summary\*\*

synth\_summary <- summary(synth\_model)

capture.output(print(synth\_summary), file = "1cResults\_Model\_Summary.txt")

# ✅ \*\*Save Result 3: Extract ATT\*\*

write.csv(synth\_summary$att, "1cResults\_ATT.csv", row.names = FALSE)

# ✅ \*\*Save Result 4: ATT Plot (All Levels)\*\*

png("1cResults\_ATT\_Plot.png", width=800, height=600)

plot(synth\_summary)

dev.off()

# ✅ \*\*Compute RMSE for Pre-Treatment Fit\*\*

pre\_treatment\_periods <- synth\_summary$att[synth\_summary$att$Time < 0, ]

pre\_treatment\_rmse <- sqrt(mean(pre\_treatment\_periods$Estimate^2, na.rm = TRUE))

# ✅ \*\*Save Result 5: Pre-Treatment RMSE\*\*

write.csv(data.frame(Pre\_Treatment\_RMSE = pre\_treatment\_rmse), "1cResults\_PreTreatment\_RMSE.csv", row.names = FALSE)

# ✅ \*\*Save Result 6: ATT Plot (Average Levels)\*\*

png("1cResults\_ATT\_Plot\_Average.png", width=800, height=600)

plot(synth\_summary, levels = "Average")

dev.off()

# ✅ \*\*Compute Pre-Treatment Fit for Each Unit\*\*

pre\_treatment\_df <- synth\_summary$att[synth\_summary$att$Time < 0, ] %>%

drop\_na(Estimate)

pre\_treatment\_rmse\_by\_unit <- pre\_treatment\_df %>%

group\_by(Level) %>%

summarise(RMSE = sqrt(mean(Estimate^2, na.rm = TRUE)))

# ✅ \*\*Save Result 7: Pre-Treatment RMSE for Each Unit\*\*

write.csv(pre\_treatment\_rmse\_by\_unit, "1cResults\_PreTreatment\_RMSE\_By\_Unit.csv", row.names = FALSE)

# ✅ \*\*Run Placebo Test\*\*

placebo\_model <- multisynth(Violent\_per\_100k ~ Treat | Female.Veterans + Population.Density + Gini.Index..Income.Inequality.,

unit = City,

time = Date,

data = analysis\_df,

permute = TRUE)

placebo\_summary <- summary(placebo\_model)

# ✅ \*\*Save Result 8: Placebo Test Summary\*\*

capture.output(print(placebo\_summary), file = "1cResults\_Placebo\_Summary.txt")

# ✅ \*\*Save Placebo Plot\*\*

png("1cResults\_Placebo\_Plot.png", width=800, height=600)

plot(placebo\_summary)

dev.off()

# ✅ \*\*Save Result 9: Individual Unit ATT Plots\*\*

png("1cResults\_ATT\_Individual\_Units.png", width=800, height=600)

plot(synth\_summary, levels = unique(analysis\_df$City))

dev.off()

# ✅ \*\*Extract Weights and Save\*\*

synth\_weights\_df <- as.data.frame(synth\_model$weights)

synth\_weights\_df$Donor\_City <- rownames(synth\_weights\_df)

synth\_weights\_long <- pivot\_longer(

synth\_weights\_df,

cols = -Donor\_City,

names\_to = "Treated\_Unit",

values\_to = "Weight"

)

treated\_units <- unique(analysis\_df$City[analysis\_df$Treat == 1])

treated\_unit\_mapping <- data.frame(

Treated\_Unit = unique(synth\_weights\_long$Treated\_Unit),

Treated\_City = treated\_units

)

synth\_weights\_named <- left\_join(synth\_weights\_long, treated\_unit\_mapping, by = "Treated\_Unit") %>%

select(Donor\_City, Treated\_City, Weight)

synth\_weights\_filtered <- synth\_weights\_named %>%

filter(Weight > 0) %>%

arrange(Treated\_City, desc(Weight))

# ✅ \*\*Save Result 10: Synthetic Control Weights\*\*

write.csv(synth\_weights\_filtered, "1cResults\_Synthetic\_Weights.csv", row.names = FALSE)

# ✅ \*\*Plot Observed vs. Synthetic Fit and Save\*\*

observed\_vs\_synth\_plot <- ggplot(synth\_summary$att, aes(x = Time, y = Estimate, group = Level)) +

geom\_line(aes(color = Level), size = 1) +

geom\_point(aes(color = Level), size = 2) +

labs(

title = "Observed vs. Synthetic Control",

x = "Time Since Treatment",

y = "Difference (Observed - Synthetic)",

color = "Treated Unit"

) +

theme\_minimal()

# Save plot

ggsave("1cResults\_Observed\_vs\_Synthetic\_Fit.png", plot = observed\_vs\_synth\_plot, width = 8, height = 6)

print("✅ All tasks completed. Results saved successfully.")

#################################PLOT THE SYNTHETIC UNIT WEIGHTS

############################# ---- PLOT 1: Improved Bar Plot for Donor Weights ----

# Create a directory to store individual plots (if it doesn’t exist)

if (!dir.exists("1cSynthetic\_Weights\_Plot")) {

dir.create("1cSynthetic\_Weights\_Plot")

}

# Filter out donor cities that have zero weights for all treated cities

synth\_weights\_filtered <- synth\_weights\_named %>%

filter(Weight > 0) %>%

group\_by(Treated\_City) %>%

filter(sum(Weight) > 0) %>% # Ensures cities with no weights at all are dropped

arrange(Treated\_City, desc(Weight)) %>%

ungroup()

# ✅ \*\*Plot for Each Treated City Individually\*\*

for (treated\_city in unique(synth\_weights\_filtered$Treated\_City)) {

plot\_data <- synth\_weights\_filtered %>% filter(Treated\_City == treated\_city)

# Create a bar plot for each treated city separately

p <- ggplot(plot\_data, aes(x = reorder(Donor\_City, Weight), y = Weight, fill = Donor\_City)) +

geom\_bar(stat = "identity") +

coord\_flip() +

labs(title = paste("1cSynthetic Weights for", treated\_city),

x = "Donor Cities", y = "Weight") +

theme\_minimal() +

theme(legend.position = "none", axis.text.y = element\_text(size = 8))

# Save each plot

ggsave(filename = paste0("1cSynthetic\_Weights\_", gsub(" ", "\_", treated\_city), ".png"), plot = p, width = 10, height = 6)

}

############VARY NU

# Define a sequence of nu values

#nu\_values <- unique(c(seq(0, 1, by = 0.05))

nu\_values <- seq(0, 1, by = 0.05)

# Initialize a dataframe to store ATT results

att\_results <- data.frame(Nu = numeric(), ATT\_Estimate = numeric(), ATT\_Std\_Error = numeric())

# Loop over different nu values

for (nu in nu\_values) {

# Run the multisynth model with a fixed nu

synth\_model <- multisynth(

Violent\_per\_100k ~ Treat | Female.Veterans + Population.Density + Gini.Index..Income.Inequality.,

unit = City,

time = Date,

data = analysis\_df,

nu = nu # Fix nu at each iteration

)

# Extract the ATT estimate and standard error (Ensure single value)

synth\_summary <- summary(synth\_model)

att\_value <- tail(synth\_summary$att$Estimate[synth\_summary$att$Level == "Average"], 1)

att\_std\_error <- tail(synth\_summary$att$Std.Error[synth\_summary$att$Level == "Average"], 1)

# Ensure we are storing only valid values

if (length(att\_value) == 1 && length(att\_std\_error) == 1 && !is.na(att\_value) && !is.na(att\_std\_error)) {

att\_results <- rbind(att\_results, data.frame(Nu = nu, ATT\_Estimate = att\_value, ATT\_Std\_Error = att\_std\_error))

} else {

print(paste("Skipping nu =", nu, "due to missing ATT or Std. Error"))

}

}

print(att\_results)

if (nrow(att\_results) > 0) {

write.csv(att\_results, "1cSimulation\_ATT\_vs\_Nu.csv", row.names = FALSE)

} else {

print("No ATT results to save. Check the loop or model output.")

}

# Load required libraries

library(ggplot2)

# Define Y-axis limits to zoom in (adjust as needed)

y\_min <- min(att\_results$ATT\_Estimate - att\_results$ATT\_Std\_Error, na.rm = TRUE)

y\_max <- max(att\_results$ATT\_Estimate + att\_results$ATT\_Std\_Error, na.rm = TRUE)

# Create the ATT vs. Nu plot with zoomed-in Y-axis

ggplot(att\_results, aes(x = Nu, y = ATT\_Estimate)) +

geom\_line(color = "black") + # Line showing ATT variation

geom\_ribbon(aes(ymin = ATT\_Estimate - ATT\_Std\_Error,

ymax = ATT\_Estimate + ATT\_Std\_Error),

fill = "gray", alpha = 0.5) + # Confidence interval (shaded)

geom\_point(color = "red", size = 3) + # Highlight individual points

labs(title = "Effect of Varying Nu on ATT Estimate",

x = expression(nu),

y = "ATT Estimate") +

theme\_minimal() +

ylim(y\_min, y\_max) # 🔥 Zoom in on Y-axis

# Save the plot with high resolution

ggsave("1cSimulation\_ATT\_vs\_Nu.png", width = 7, height = 5, dpi = 300)